ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ ОПРЕДЕЛЕНИЯ ДЛИНЫ КЛЮЧА И ДЕШИФРОВАНИЯ ПЕРЕСТАНОВОЧНОГО ШИФРА

Алексей Городилов, Владимир Морозенко

***Аннотация**: В статье рассматривается возможность применения генетических алгоритмов к задачам криптоанализа. Разработан и описан генетический алгоритм для поиска секретного ключа блочного перестановочного шифра. Ключом в данном случае является перестановка начального фрагмента натурального ряда. Разработанный алгоритм точно определяет длину секретного ключа и с регулируемой «точностью» находит саму секретную перестановку. Анализ результатов вычислительного эксперимента свидетельствует о возможности почти полного автоматического дешифрования текста.*

***Keywords**: криптография, криптоанализ, блочный перестановочный шифр, генетический алгоритм, защита информации.*

***ACM Classification Keywords**: I.2 Artificial Intelligence: I.2.8 Problem Solving, Control Methods, and Search - Heuristic methods.*

Введение

Основными задачами криптологии являются разработка надежных схем шифрования (задача криптографии) и нахождение эффективных методов дешифрования существующих схем (задача криптоанализа). Криптографический способ защиты информации предусматривает такое её преобразование, при котором она становится доступной для прочтения лишь обладателю секретного ключа. Надежность этого способа защиты определяется стойкостью используемой схемы шифрования к криптоанализу. При криптоанализе конкретного шифра предполагается, что сама схема шифрования известна, а неизвестным является только секретный ключ и/или его длина. Иными словами, задача вскрытия шифра состоит в нахождении единственного настоящего секретного ключа среди множества всех возможных ключей, т.е. является задачей поиска. При этом пространство поиска велико, а критерий «качества» найденного решения, как правило, не поддается строгой формализации.

В данной работе рассматривается задача криптоанализа блочного перестановочного шифра. Секретным ключом здесь является перестановка начального фрагмента натурального ряда, длина которой также неизвестна. Для решения поставленной задачи в данной работе применяются генетические алгоритмы. Поскольку одной из областей успешного применения генетических алгоритмов являются именно задачи оптимизации и поиска, то их использование в данном случае представляется вполне объяснимым. Вопросам применения генетических алгоритмов в криптологии посвящен ряд статей [Delman, 2005, Лебедев, 2005, Jakobsen, 1995], но в них длина ключа считается известной, используются стандартные подходы, не учитывающие специфику задачи, либо не исследуется влияние параметров на скорость сходимости алгоритмов.

Разработке всякого генетического алгоритма должны предшествовать выбор подходящего способа кодирования допустимого решения в виде символьной строки, а также описание основных операторов – селекции, кроссовера и мутации. Качество алгоритма должно регулироваться за счет вариации его параметров – численности популяции, числа поколений, вероятностных характеристик основных операторов.

Блочный перестановочный шифр

В данной работе рассматривается конкретная схема шифрования – симметричный блочный перестановочный шифр. Она состоит в том, что входной текст разбивается на блоки, т.е. строки фиксированной длины N, а затем внутри каждого блока символы переставляются в соответствии с заданной перестановкой

$$P = \begin{pmatrix} 1 & 2 & 3 & \dots & i & \dots & N \\ p_1 & p_2 & p_3 & \cdots & p_i & \cdots & p_N \end{pmatrix}.$$

где $p_i \in \{1,2,3,\dots,N\}$. Иными словами, символ, стоящий на i-й позиции, перемещается на позицию p_i. Секретным ключом в таком шифре является перестановка P. Расшифровывание происходит с использованием обратной перестановки P^{-1}.

Генетический алгоритм

Поскольку решением задачи криптоанализа в данном случае является перестановка, то и особью в генетическом алгоритме будем считать перестановку. Для начала предположим, что длина ключа нам известна, и нам остается лишь найти сам ключ, т.е. перестановку фиксированной длины N.

Важный вопрос, который необходимо решить – какой смысл будет вкладываться в отдельные гены особи. Простейший вариант, который является на первый взгляд наиболее очевидным, – считать отдельными генами элементы перестановки P, то есть i-ым геном особи считать число p_i. Очевидно, что при таком подходе гены получаются зависимыми друг от друга. Если какой-то ген равен j, то никакой другой ген этой особи уже не должен принимать значение j, так как в перестановке P все числа от 1 до N встречаются ровно по одному разу. Зависимость генов накладывает значительные ограничения на операторы мутации и скрещивания. Стандартные операторы в данной ситуации неприменимы, так как все они работают с представлением набора генов в виде строки независимых бит. Такие зависимости генов не свойственны живой природе, что делает аналогию неточной и ставит под сомнение эффективность применения генетического алгоритма. Тем не менее, выбранная интерпретация генов как элементов перестановки интуитивно понятна и не требует дополнительных затрат на их формирование.

Заметим, что альтернативным подходом могло бы стать использование промежуточного представления особей в виде некоторого объекта, легко трансформируемого в перестановку. В то же время такой объект должен задаваться при помощи битовой строки, так чтобы были применимы стандартные операторы скрещивания и мутации. При этом подходе задача выбора подходящего промежуточного представления особи может оказаться достаточно трудной. В данной работе выбран первый – интуитивно понятный – из указанных подходов, т.е. везде в дальнейшем в качестве отдельных генов будут рассматриваться элементы перестановки.

Следующий вопрос – как вычислять приспособленность особей. Мы будем исходить из предположения, что шифруемый текст представляет собой осмысленный текст на русском языке. Фитнесс-функция (целевая функция) будет заимствована из работ Якобсена [Jakobsen, 1995, Аграновский, 2002]. Томас Якобсен в 1995 году предложил автоматический метод раскрытия ключа шифра простой замены. В своем методе вскрытия шифров замены Якобсен использовал информацию о распределении частот встречаемости биграмм в осмысленных текстах. Биграмма – это две подряд идущие буквы в тексте. Целевую функцию Якобсен предложил вычислять как сумму модулей разностей между заранее известным среднестатистическим количеством биграмм в осмысленных текстах и их реальным

количеством в шифртексте. Пусть T_{ij} – это относительные частоты встретившихся в тексте T биграмм (i j). Тогда целевая (фитнесс-) функция будет иметь вид

$$W(T) = \sum_{ij} \left| T_{ij} - E_{ij} \right|,$$

где E_{ij} – относительные частоты биграмм, заранее известные и зафиксированные в алгоритме в качестве эталонных значений. Матрица частот E высчитывается заранее на осмысленных текстах большой длины, т.е. отражает среднестатистическое распределение биграмм. Нетрудно видеть, что чем ближе текст к осмысленному, тем меньше значение целевой функции и тем «ближе» найденный ключ к настоящему секретному ключу. Это означает, что меньшему значению целевой функции соответствует большее значение приспособленности особи, и наоборот. Важно подчеркнуть, что указанная фитнесс-функция W, вообще говоря, не обращается в ноль, даже если расшифрованный текст T получен из шифр-текста при использовании настоящего секретного ключа.

Таким образом, если нам дан зашифрованный текст S, то для вычисления приспособленности особи P, необходимо выполнить следующие шаги.

1. Расшифровать зашифрованный текст S с использованием выбранного ключа P, в результате чего получим текст $T = Decrypt_P(S)$.
2. Подсчитать частоты T_{ij} всевозможных биграмм (i j) в тексте T.
3. Найти значение целевой функции $W(T)$ по указанной выше формуле.

Более подробно рассмотрим свойства фитнесс-функции $W(T)$. Пусть две особи P_1 и P_2 различаются только двумя первыми генами (очевидно, что только одним геном они отличаться не могут). Положим для определенности

$$P_1 = \begin{pmatrix} 1 & 2 & 3 & \dots & N \\ p_1 & p_2 & p_3 & \cdots & p_N \end{pmatrix},$$

$$P_2 = \begin{pmatrix} 1 & 2 & 3 & \dots & N \\ p_2 & p_1 & p_3 & \cdots & p_N \end{pmatrix}.$$

Пусть имеется зашифрованное сообщение S. Тогда расшифрованные при помощи ключей P_1 и P_2 тексты $Decrypt_{P_1}(S)$ и $Decrypt_{P_2}(S)$ отличаются только символами, стоящими на позициях с номерами p_1, p_2, $p_1 + N$, $p_2 + N$, $p_1 + 2N$, $p_2 + 2N$ и т.д. Номера указанных позиций образуют две арифметических последовательности с разностью N. Поскольку тексты $Decrypt_{P_1}(S)$ и $Decrypt_{P_2}(S)$ могут отличаться только в указанных позициях, то в каждом блоке длины N этих текстов не более трех биграмм будут отличаться своими частотами. Таким образом, при достаточно большой длине N ключа, что справедливо для реальных систем, небольшому изменению особи будет соответствовать небольшое изменение целевой функции.

Заметим также, что в достаточно длинных осмысленных текстах частоты встречаемости биграмм T_{ij} близки к соответствующим среднестатистическим вероятностям E_{ij}. Поэтому в результате дешифровки текста S с помощью настоящего секретного ключа значение целевой функции должно быть близким к нулю. Значит, если K – настоящий секретный ключ, то величина $W(Decrypt_K(S))$ должна быть минимально возможной.

Как было сказано выше, стандартные операторы скрещивания применимы только тогда, когда особь представляется битовой строкой, состоящей из независимых бит. Поскольку в нашем случае это не так, то

стандартные операторы неприменимы и требуется разработка оператора кроссовера (скрещивания) специального вида. Основное требование, накладываемое на данный оператор, заключается в том, чтобы в результате его применения всегда получалась допустимая перестановка. В данной работе предлагается следующий оператор скрещивания.

1. Гены пронумерованы числами 1, 2, 3, …, N и просматриваются в порядке возрастания номеров.
2. Ген с очередным номером берется от одного из предков, если это возможно.
3. Если гены от обоих родителей недопустимы, берется произвольное допустимое число.

Будем использовать обозначение $P(i)$ для числа, на которое перестановка P заменяет число i. Таким образом, получаем следующий алгоритм скрещивания двух родителей P_1 и P_2 и получением потомка P^*.

1. Множеству *Used* уже использованных значений генов присвоить начальное значение Ø, установить номер текущего вычисляемого гена $i = 1$.
2. Выяснить, принадлежат ли числа $P_1(i)$ и $P_2(i)$ множеству *Used*.
3. Если множеству *Used* не принадлежит только одно из чисел $P_1(i)$ и $P_2(i)$, то присвоить его i-му гену потомка P^*. Если множеству *Used* не принадлежат оба числа $P_1(i)$ и $P_2(i)$, то либо с вероятностью p_c выбрать число $P_1(i)$, либо с вероятностью $(1 – p_c)$ выбрать $P_2(i)$. Выбранное число присвоить i-му гену потомка P^*. Наконец, если оба числа лежат во множестве *Used*, то i-му гену потомка P^* присвоить произвольное число из множества {1, 2, 3, …, N}\ *Used.*
4. Число, которое было присвоено i-му гену потомка P^*, включить во множество *Used*.
5. Перейти к рассмотрению следующего гена, т.е увеличить i на единицу и перейти к шагу 2.

В качестве оператора мутации можно использовать стандартный оператор обмена, при котором с заданной вероятностью в особи меняются местами два гена. При селекции особей использовалось две основные классические идеи: «принцип рулетки» и «принцип элитизма» [Mitchell, 1999]. Размер популяции оставался постоянным за счет того, что каждый раз при появлении двух потомков, из популяции удалялись две наименее приспособленные особи. Наконец, условием окончания работы предлагаемого генетического алгоритма было выбрано превышение количества эпох заранее фиксированной величины M. Это позволяет заранее предсказать время, которое потребуется для работы алгоритма или, наоборот, задать параметр M так, чтобы алгоритм завершил свою работу за указанное ограниченное время.

Влияние параметров генетического алгоритма

Итак, выше был предложен генетический алгоритм для решения задачи криптоанализа блочного перестановочного шифра в предположении, что длина N ключа известна. Исследуем вопрос об оптимальных значениях параметров этого алгоритма. Выделим те параметры алгоритма, которые упоминались в тексте и которые оказывают влияние на скорость сходимости и качество получаемого решения:

- численность начальной популяции k;
- вероятность унаследования генов от одного из родителей p_c;
- вероятность мутации p_m;
- количество поколений M.

Каких-либо общих теоретических рекомендаций для выбора этих параметров не существует. Можно лишь сказать, что вероятность мутации должна быть незначительной, а вероятность унаследования генов от одного из родителей, наоборот, должна быть величиной, близкой к 0,5. Количество поколений может быть выбрано исходя из имеющихся временных ресурсов. Вообще говоря, если единственным условием

окончания работы алгоритма является количество поколений, то оно должно быть достаточно большим. Также важна численность начальной популяции, поскольку в предлагаемом генетическом алгоритме численность популяции остается неизменной и всегда равна *k*. Следует отметить, что численность популяции существенным образом влияет на объем вычислений, и, следовательно, на затрачиваемое время.

Проведенные численные эксперименты показали, что малая численность популяции дает плохие результаты: алгоритм быстро находит точку локального минимума фитнесс-функции, и все дальнейшие популяции формируются в его окрестности. Оказалось, что при длине ключа, равной 10, оптимальным значением численности популяции является 15-17 особей. Изменения показателя вероятности мутации в разумных пределах не оказали заметного влияния на работу алгоритма. Частично это можно объяснить тем, что специфически выбранный оператор скрещивания уже подразумевает некоторую мутацию. Действительно, когда ген не может быть унаследован ни от одного из предков, он просто выбирается случайным образом, то есть в этом гене потомок получается «непохожим» ни на одного из своих предков. Оптимальное число поколений сильно зависит от длины *N* ключа. При малых значениях параметра *N* слишком большое число поколений неэффективно, так как после некоторого числа поколений алгоритм находит локальный минимум, и дальнейшие действия не приводят к улучшению решения. С ростом длины ключа быстро растет пространство возможных решений и, естественно, требуется большее число поколений для нахождения наилучшего решения.

Получаемый в результате работы алгоритма текст далеко не всегда совпадает с исходным открытым текстом, однако не сильно от него отличается. Во многих случаях, восстановить отдельные символы можно с помощью контекста. Таким образом, алгоритм хотя и не решает задачу криптоанализа полностью автоматически, но значительно помогает дешифровать зашифрованный текст.

Определение длины ключа

Описанный выше генетический алгоритм работает при известной заранее длине ключа *N*. Результатом его работы является перестановка, соответствующая наиболее приспособленной особи из финальной популяции. Поскольку эта перестановка доставляет целевой функции *W*(*T*) наименьшее из её найденных значений, то побочным эффектом в работе описанного алгоритма является вычисление по заданной длине ключа *N* числа *F*(*N*), указывающего на минимальное найденное значение целевой функции.

Рассмотрим теперь задачу нахождения длины ключа по имеющемуся зашифрованному тексту. Пусть *N* – предполагаемое значение длины настоящего секретного ключа, а *L* – его настоящая длина. Если числа *N* и *L* совпадают, то можно ожидать, что найденная в результате работы генетического алгоритма перестановка будет близка к искомой, а целевая функция при использовании этой перестановки примет относительно небольшое значение.

Пусть настоящим секретным ключом является перестановка

$$K = \begin{pmatrix} 1 & 2 & \cdots & L \\ p_1 & p_2 & \cdots & p_L \end{pmatrix}.$$

Рассмотрим ключ длины $2 \cdot L$ вида

$$K' = \begin{pmatrix} 1 & 2 & \cdots & L & L+1 & \cdots & 2 \cdot L \\ p_1 & p_2 & \cdots & p_L & L+p_1 & \cdots & L+p_L \end{pmatrix},$$

в котором первые *L* столбцов в точности совпадают с первыми *L* столбцами перестановки *K*, а остальные получены из первых *L* столбцов перестановки *K* увеличением всех чисел на *L*. Поскольку оба ключа K и K' превращают любой исходный текст в идентичные шифр-тексты, то шифр-текст будет правильно расшифрован также и при помощи ключа K'. Это означает, что при поиске ключа длины 2·*L* генетический алгоритм найдет перестановку, близкую к ключу K'. При этом найденной перестановке будет соответствовать относительно небольшое значение целевой функции, поэтому полученная на выходе алгоритма величина *F*(2·*L*) также примет небольшое значение. То же самое справедливо для всех ключей с длинами, кратными числу *L*. Иными словами, если на вход генетическому алгоритму подать шифр-текст и длину ключа *N* = *m*·*L*, где *m* – натуральное число, то на выходе алгоритма следует ожидать перестановку с небольшим значением фитнесс-функции. Если же *N* не кратно *L*, то перестановок длины *N*, близких к ключу *K*, вообще не существует. В этом случае полученное значение фитнесс-функции и, соответственно, величина *F*(*N*), должны быть относительно велики. Таким образом, для всех *N*, не кратных *L*, величина *F*(*m*·*L*) должна быть существенно меньше, чем *F*(*N*).

Исходя из вышесказанного, можно предложить следующий алгоритм определения длины ключа. Выбирается некоторое начальное значение длины ключа N_0 и для него с помощью имеющегося генетического алгоритма вычисляется значение $F(N_0)$. На следующем шаге предполагаемую длину ключа увеличиваем на единицу и, вновь применив генетический алгоритм, вычислим $F(N_1)$, где $N_1 = N_0 + 1$, и т.д. На *i*-ом шаге вычисляем значение $F(N_i)$, где $N_i = N_0 + i$. Увеличение длины ключа производятся до достижения некоторой установленной заранее верхней границы *N**. В результате такого итерационного процесса, на каждом шаге которого запускается описанный генетический алгоритм, образуется последовательность чисел

$$F(N_0),\ F(N_0 + 1),\ F(N_0 + 2),\ \ldots,\ F(N_0 + i),\ \ldots,\ F(N^*).$$

В ней своими малыми значениями должны выделяться элементы, чьи номера образуют арифметическую прогрессию с шагом *L*. Величина шага *L* и будет являться искомой длиной настоящего секретного ключа.

Отметим, что для ускорения работы алгоритма при нахождении длины ключа можно использовать малые численности популяции и небольшое число поколений, а после нахождения истинной длины ключа *L* можно повторно применить генетический алгоритм с большими значениями параметров для отыскания самого секретного ключа.

Пример работы алгоритма

В качестве открытого текста был взят фрагмент литературного текста на русском языке – отрывок из рассказа И.С. Тургенева «Рудин»:

> *Было тихое летнее утро. Солнце уже довольно высоко стояло на чистом небе, но поля еще блестели росой, из недавно проснувшихся долин веяло душистой свежестью, и в лесу еще сыром и не шумном, весело распевали ранние птички. На вершине пологого холма, сверху донизу покрытого…*

В вычислительном эксперименте этот текст был зашифрован при помощи блочного перестановочного шифра с длиной ключа, равной 10. В результате был получен следующий набор символов:

> *т иолоеыхБенеуеттл оСлц. еонрд оожелув ьвско оноьляонтоас отм ис ч,н бепеонщеебя лл оил отессрези е, дйноп рсноновоасхядшиов уевяон иллтсо шисуйдтсь,же еювелс в е уирыо сиемщушмое мнн есл вер о,авл пе сиаеи тнниапрН ави.ек ч еплиношорх омгоаолгрех свд у, упкизрноо о ог т ы…*

Исходя из такого бессмысленного набора символов, трудно воссоздать исходный текст «вручную». Далее этот зашифрованный текст был подан на вход разработанного генетического алгоритма с целью определения длины секретного ключа. Диапазон предполагаемых значений длины ключа составил [4; 34], т.е. были выбраны параметры $N_0 = 4$, $N^* = 34$. Для каждой предполагаемой длины ключа от 4 до 34 были найдены соответствующие значения $F(4)$, $F(5)$, …, $F(34)$. На рис. 1 для удобства представлены обратные значения $1/F(4)$, $1/F(5)$, …, $1/F(34)$. Чем выше точка на графике, тем ближе её абсцисса к истинной длине L секретного ключа или к кратной ей величине $m \cdot L$.

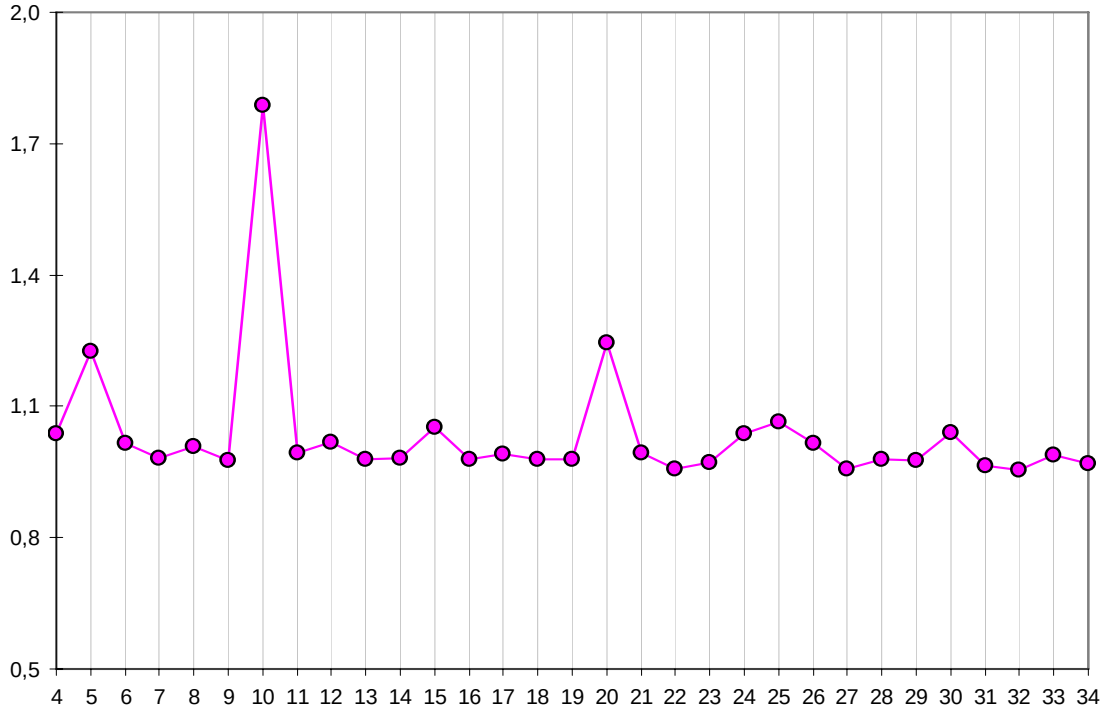

Рис. 1. График зависимости 1/F(N), где N – предполагаемая длина секретного ключа

На рисунке виден максимальный «всплеск» функции приспособленности, который приходится на значение длины, равное 10, что соответствует длине L настоящего секретного ключа. Следующий заметный локальный максимум наблюдается при $N = 20$, что равно удвоенному значению $2 \cdot L$. Далее заметных «всплесков» не наблюдается, хотя при $N = 30$ (т.е. при утроенном значении $3 \cdot L$) можно заметить локальный максимум.

Результаты вычислительного эксперимента хорошо согласуется с теоретическими предсказаниями. Постепенное затухание «всплесков» функции приспособленности по мере увеличения параметра N связано с тем, что для всех значений N генетический алгоритм работал с одним и тем же набором параметров. В соответствии же с вышесказанным, для больших значений предполагаемой длины ключа требуется увеличивать количество поколений. В противном случае, при малом числе поколений генетический алгоритм просто «не успевает» найти хорошее решение.

Таким образом, в данном эксперименте на основе анализа полученных данных можно утверждать, что длина секретного ключа равна $L = 10$. Чтобы теперь найти секретный ключ, повторно был применен генетический алгоритм со следующим набором параметров: численность популяции – 15, количество поколений – 30, вероятность мутации – 0,2. В результате работы алгоритма был получен следующий текст:

Былохти ое лет еенутро. нолСце ужевдо ольно оысвко сто лояна чис омт небе он, поля щееблестери лосой, нз иедавноопр снувши сяхдолин леяво душийтос свежеютьс, и в уесл еще смроы и не нумшом, веоелс распеиалв раннпе итички. На вершино пелогоголхо ма, свурхе донизо пукрытог о…

Как видно, полностью дешифровать зашифрованный текст не удалось, т.к. полученный текст не совпадает с исходным. Однако можно заметить, что он «близок» к исходному. Например, в нем можно заметить осмысленные слова, которые совпадают с соответствующими словами исходного текста: *было*, *уже*, *поля*, *небе и др.* На основе полученного текста уже можно «вручную» довести процесс дешифрования до конца.

Кроме того, секретный ключ, которым в данном случае является перестановка

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 10 & 8 & 5 & 6 & 2 & 1 & 3 & 9 & 4 & 7 \end{pmatrix},$$

и ключ, полученный в результате работы генетического алгоритма

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 10 & 8 & 5 & 6 & 9 & 1 & 3 & 2 & 4 & 7 \end{pmatrix},$$

отличаются только в одной паре значений, что в генетическом алгоритме соответствует одной паре хромосом. Учитывая, что исходный открытый текст был осмысленным, нетрудно «вручную» перебором небольшого числа вариантов найти настоящий секретный ключ.

Безусловно, предложенный в данной работе генетический алгоритм поиска секретного ключа хотя и не автоматизирует полностью процесс дешифровки, но довольно существенно ускоряет его, делая несложным доведение «вручную» процесс дешифровки до получения осмысленного текста.

Библиографический список

[Delman, 2005] Delman B. GeneticAlgorithms in Cryptography // A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering. New York, 2004.

[Лебедев, 2005] Лебедев А. В криптографии мы способны конкурировать / ЛАН Крипто, 2005.

[Jakobsen, 1995] Jakobsen T. A Fast Method for the Cryptanalysis of Substitution Ciphers, 1995.

[Аграновский, 2002] Аграновский А. В., Хади Р.А. Практическая криптография: алгоритмы и их программирование. М.:СОЛОН-Пресс, 2002.

[Mitchell, 1999] Mitchell M. An Introduction to Genetic Algorithms. Fifth printing. Cambridge, MA: The MIT Press, 1999.

Сведения об авторах

Алексей Городилов – Пермский государственный университет, студент магистратуры кафедры математического обеспечения вычислительных систем; Россия, г. Пермь, 614990, ул. Букирева, д. 15; e-mail: gora830@yandex.ru

Владимир Морозенко – Пермский государственный университет, доцент кафедры математического обеспечения вычислительных систем; Россия, г. Пермь, 614990, ул. Букирева, д. 15; e-mail: v.morozenko@mail.ru