УДК 004.056.55

Биоинспирированные методы криптоанализа асимметричных алгоритмов шифрования на основе факторизации составных чисел

А. С. Сергеев
(Донской государственный технический университет),
О. П. Третьяков, А. Е. Васильев
(филиал Военной академии связи),
Ю. О. Чернышёв
(Донской государственный технический университет)

Рассматривается возможность применения биоинспирированных методов для решения задачи криптоанализа асимметричных алгоритмов шифрования на основе факторизации составных чисел. Представлены алгоритмы муравьиных и пчелиных колоний для разложения составных чисел на множители путём определения делителя числа с заданной точностью в заданном интервале. Описаны отличительные особенности представленных методов, в том числе возможность эффективной параллельной реализации.
Ключевые слова: *криптоанализ, пчелиный алгоритм, муравьиный алгоритм, феромон, факторизация числа, асимметричные криптосистемы, биоинспирированные методы.*

Введение. В настоящее время в науке и технике находят широкое применение алгоритмы, основанные на природных системах. Это новое научное направление под названием «природные вычисления» объединяет математические методы, в которых заложен принцип природных механизмов принятия решений. К ним относятся методы моделирования отжига, генетические, эволюционные методы, алгоритмы роевого интеллекта и др. [1].

Однако, как отмечено в [2], недостатком эволюционных методов является использование «слепого» поиска, что в общем случае приводит к следующим проблемам: генерация решений с нарушениями и, как следствие, увеличение времени поиска и необходимость дополнительного контроля; генерация большого количества одинаковых решений; генерация большого количества плохо приспособленных решений, что в общем случае может привести к попаданию в локальный оптимум. Поэтому представляет интерес применение эвристических методов, инспирированных природными системами, в которых осуществляется поэтапное построение решения задачи (т. е. добавление нового оптимального частичного решения к уже построенному частичному оптимальному решению). К методам данного вида относят муравьиные и пчелиные алгоритмы, основные идеи и принципы которых описаны, например, в [1]. Известны случаи применения алгоритмов роевого интеллекта для оптимизации широкого круга задач, в том числе в криптоанализе. Вместе с тем следует заметить, что наряду с классическими симметричными алгоритмами шифрования в настоящее время используются методы асимметричной криптографии — сравнительно молодой области науки. Один из её первых алгоритмов — RSA [3, 4], сложность которого определяется трудностью факторизации больших чисел. Как отмечено в [5], факторизация (задача разложения числа на простые множители) — одна из основных теоретико-числовых задач, используемых в криптографии.

Постановка задачи криптоанализа асимметричных криптосистем. Как отмечено в [6], во всех асимметричных криптосистемах (в т. ч. RSA) используются два ключа: один для шифрования ($K_{откр}$), другой для дешифрования ($K_{секр}$). Ключи представляют собой пары ($K_{откр}$, N), ($K_{секр}$, N), где N — модуль, при этом $N = P \cdot Q$, где P и Q — случайные большие простые числа. При этом возможны несколько вариантов криптоанализа:

1. На основе известного открытого текста и шифртекста подобрать такой секретный ключ $K_{секр}$, чтобы выполнялось равенство $M_i = C_i^{Kсекр} (\text{mod } N)$, где M_i — открытый текст, C_i — шифртекст.

2. Опытным путём определить функцию Эйлера $\varphi(N) = (P - 1)(Q - 1)$, разложив модуль N на множители P и Q, и секретный ключ $K_{секр}$ из соотношения $K_{секр} \cdot K_{откр} = 1 (\text{mod } \varphi(N))$.

3. Подобрать такие числа P и Q, чтобы выполнялось соотношение $N = P \cdot Q$.

Поскольку в варианте 1 нахождение секретного ключа $K_{секр}$ имеет комбинаторную сложность (в общем случае может потребоваться полный перебор), как и в варианте 2 (требуется полный перебор всех взаимно простых чисел в интервале $[1, \varphi(N)]$, то актуальна задача исследования возможности применения современных технологий природных алгоритмов для определения всех возможных делителей составного числа N.

Отметим, что ранее в [7] был представлен генетический алгоритм (ГА) для решения задачи определения вариантов разложения заданного числа N на множители. Однако при реализации данного алгоритма основной также является задача определения числа *Val*, являющегося делителем числа N. Для решения этой задачи в [7, 8] предлагается алгоритм, сущность которого заключается в определении на отрезке $[0, N]$ методом генетического поиска точки R, удовлетворяющей условиям $(N - R) / R = n$ — целое (т. е. $(N - R) / R - [(N - R) / R] = 0$),

$$R \cdot (n + 1) = N, \qquad (1)$$

где $[x]$ — целая часть числа x, или множества точек R, R_1, R_2, …, R_k, удовлетворяющих условиям $R_1 / R = n_1$, $(R_2 - R_1) / R = n_2$, $(R_3 - R_2) / R = n_3$, …, $(R_k - R_{k-1}) / R = n_k$, $R \times (n_1 + n_2 + \ldots + n_k) = N$, где R, n_k — целые числа. Очевидно, что выражение (1) может быть использовано в качестве целевой функции в ГА.

Однако основной проблемой при реализации данного ГА является нахождение экстремума немонотонной функции, т. е. функции, значение $f(x)$ которой в каждой точке x является, по сути, случайной величиной и не даёт информации о приближении к глобальному экстремуму. Таким образом, при нахождении экстремума функции $f(R) = (N - R) / R - [(N - R) / R]$ (т. е. значения R, для которого $f(R) = ((N - R) / R - [(N - R) / R] = 0)$) с использованием структуры ГА фактически имеет место «слепой» поиск, и, как отмечено в [2], это является основным недостатком большинства структур ГА.

На основе логарифмического закона распределения простых чисел [9] в работе [6] описана модель ГА для нахождения разложения составного числа N на 2 простых множителя. В данной модели на отрезке $[0, N]$ осуществляется генерация популяции простых чисел. Случайным выбором битов генерируется случайное n-битовое число G. Далее на отрезке $[G - \ln(G), G + \ln(G)]$ осуществляется поиск наиболее вероятного простого числа. Каждое $x \in [G - \ln(G), G + \ln(G)]$ последовательно проверяется на делимость с простыми числами в интервале $[3, 2 \cdot \ln(G)]$. Так как реальное значение G составляет порядка $G \approx 2^{512}$, то радиус поиска представляется как $r = \ln G = \log_2 G / \log_2 e = \log_2 G / 1{,}442695 = n / 1{,}442695$.

Далее к полученной популяции простых чисел применяется множество генетических операций для получения простых чисел-потомков. Для этого используется описанная методика поиска наиболее вероятного простого числа в окрестности декодированного числа. Однако, учитывая реальную размерность задачи (как отмечено в [3, 6], разработчикам криптоалгоритмов на базе RSA приходится применять числа длиной не менее 200 десятичных знаков, на данный момент криптостойким считается ключ размерностью 2^{1024}), этот алгоритм может потребовать значительных временных ресурсов (при операции формирования большой популяции простых чисел и их получении при проведении генетических операций) и оказаться достаточно трудоёмким при практической реализации. Отметим также, что при использовании известных технологий распараллеливания ГА (например, «островной» модели), описанных, в частности, в [8, 10], необходимо организовать межпроцессорные связи между «островами» (группами процессоров, моделирующих

развитие популяции), чтобы предотвратить потери хороших решений и попадание в локальный оптимум. Как отмечено в [10], частота миграции является наиболее существенным фактором и должна быть установлена на основе экспериментальных результатов, что также может потребовать временных ресурсов и снизить эффективность работы ГА.

Поэтому представляет несомненную актуальность разработка новых методов, использующих модели процессов живой природы и ликвидирующих отмеченные недостатки ГА путём отмеченного выше поэтапного построения оптимальных решений. Рассмотрим возможный подход для решения задачи разложения составного числа на простые сомножители с помощью другого класса биоинспирированных методов — алгоритмов муравьиных и пчелиных колоний.

Разработка метода муравьиных колоний для факторизации составных чисел. Основные положения теории муравьиных алгоритмов и описание их работы приводятся в [1, 11, 12, 13]. В соответствии с [1], для того, чтобы построить подходящий муравьиный алгоритм для решения задачи, необходимо представить задачу как набор компонент и переходов или как набор неориентированных взвешенных графов, на которых муравьи могут строить решения. Поэтому рассмотрим возможный подход для сведения задачи разложения составного числа на множители к задаче нахождения кратчайшего пути в графе. Пусть задано достаточно большое число N, которое необходимо проверить на простоту и определить его делители на отрезке $[n_i, n_j]$ с заданной степенью точности, т. е. найти такие числа x_i, для которых $((N / x_i) - [N / x_i]) \to \min$. Пусть числа $x_k \in [n_i, n_j]$ содержатся в оперативной памяти одного процессора (или «острова» процессоров) и являются вершинами полного графа $G = (X, U)$, где $|X| = n_j - n_i + 1$ — множество вершин, $|U| = (n_j - n_i + 1)(n_j - n_i) / 2$ — множество рёбер, при этом вес вершины x_j равен $F(x_j) = (N / x_j) - [N / x_j]$, т. е. равен дробной части частного, полученного от деления N на x_j. Сформулируем задачу нахождения кратчайшего пути в следующей форме: найти маршрут T в графе G, содержащий заданное число m вершин, удовлетворяющий условию

$$\sum_{i=1,2,\ldots m} F(x_i) \to \min$$

Вес каждого маршрута T_k определим как (L_k, F_k), где

$$L_k = \sum_{x_i \in T_k} F(x_i), \qquad F_k = \min_{x_i \in T_k} (F(x_i)).$$

Таким образом, задачей муравьиного алгоритма в данном случае является определение маршрута, имеющего минимальный вес L_k и содержащего m вершин и вершину графа x_k, для которой $F(x_k) = (N / x_k) - [N / x_k] \to \min$, т. е. которая является наиболее точным делителем числа N. Так как цикл по времени жизни колонии продолжается до тех пор, пока не будут просмотрены все вершины графа G, то вершина $x_k \in [n_i, n_j]$ будет определена за число временных циклов C алгоритма не менее

$$C = \frac{n_j - n_i + 1}{m \cdot M} \tag{2}$$

где M — размер популяции муравьёв, m — число вершин в маршруте.

Предположим, что муравьи обладают свойствами поведения (память, зрение, обоняние), описанными в [11, 12]. Так как веса вершин $F(x_j)$ априорно неизвестны и определяются в процессе работы алгоритма, то вероятность перехода муравья k из вершины x_i в x_j устанавливается на основе соотношения

$$P_{ij}(t) = \frac{\tau_{ij}(t)}{\sum_{l \in J_{ik}} \tau_{il}(t)}, \tag{3}$$

где J_{ik} — множество вершин, доступных муравью k, находящемуся в вершине i, $\tau_{ij}(t)$ — уровень феромона в момент времени t на ребре $x_i x_j$, t — параметр цикла времени жизни колонии.

После прохождения k-м муравьём маршрута из m вершин подсчитывается длина пути. Она равна сумме всех весов вершин, по которым прошёл муравей. На каждом ребре $x_i x_j$ маршрута количество отложенного феромона составляет

$$\Delta\tau_{ij}^{k}(t) = Q / L_k(t),$$

где Q — параметр порядка длины оптимального пути (определяющий уменьшение $\Delta\tau_{ij}^{k}$ с увеличением длины маршрута $L_k(t)$).

Обновление феромона производится в соответствии с выражением

$$\tau_{ij}(t+1) = (1-\rho)\cdot\left(\tau_{ij}(t) + \sum_{k\in F}\Delta\tau_{ij,k}(t)\right) \qquad (4)$$

где F — множество муравьёв, использовавших в маршруте ребро $x_i x_j$, ρ — интенсивность испарения (в [13] предлагается выбирать $\rho = 0{,}6$).

Отметим, что в соответствии с [12] на начальном этапе задаётся начальное расположение муравьиной колонии. В общем случае для данной задачи может быть использована стратегия «дробовика», когда ограниченное количество агентов M случайным образом размещается в вершинах графа без повторений. При такой стратегии в оптимальном случае число циклов жизни колонии C определяется выражением (2).

Итак, муравьиный алгоритм для задачи нахождения делителей числа N можно сформулировать следующим образом.

1. Задание начальных значений параметров Q, M, ρ, $t_{макс}$.
2. Определение минимальных весов рёбер на основе начальной концентрации феромона.
3. Задание оценки L^* — длина кратчайшего маршрута, оценки F^* — точность решения.
4. Цикл по времени жизни колонии: $t = 1$.
5. Размещение популяции M муравьёв в случайные вершины графа без совпадений.
6. Цикл по муравьям: $k = 1$.
7. Муравей k строит маршрут $T_k(t)$ длиной $L_k(t)$ на основе распределения вероятности по рёбрам в соответствии с формулой (3).
8. $k = k + 1$, если $k \le M$, переход к 7.
9. Проверка всех маршрутов T_k на лучшее решение по сравнению с L^* и F^*.
10. В случае, если получен маршрут $T_k(t)$ длиной $L_k(t)$, которая предпочтительнее L^*, и маршрут содержит вершину x_j, для которой $F(x_j) < F^*$, обновить L^* и F^*.
11. Цикл по рёбрам графа $i = 1$.
12. $j = 1$.
13. Обновить неминимальные следы феромона на ребре x_{ij} в соответствии с (4).
14. $j = j + 1$, если $j \le n_j - n_i + 1$, перейти к 13.
15. $i = i + 1$, если $i \le n_j - n_i$, перейти к 12.
16. $t = t + 1$, если условия остановки не выполнены, переход к 5.
17. Вывести маршрут кратчайшей длины L_k, а также вершину x, для которой $F(x)$ минимально.

Условиями остановки могут быть $t > t_{макс}$, прекращение минимизации длины маршрутов, определение вершины x, для которой $F(x) = 0$.

Таким образом, данный алгоритм позволяет определять маршрут, содержащий заданное число вершин x_i, для которых $\sum F(x_i) = \min$ (т. е. являющихся самыми точными делителями числа N на отрезке $[n_i, n_j]$), а также найти вершину x_i, для которой $F(x_i) = \min$, т. е. которая является делителем числа N с заданной степенью точности.

Рассмотрим пример. Пусть $N = 893$ и в памяти процессора (или «острова» процессоров) содержится отрезок [5, 45], т. е. в этом случае $|X| = 41$. Определим $M = 4$ (количество муравьёв),

m = 4 (число вершин в маршруте), Q = 4. Начальную концентрацию феромона при t = 0 зададим равной $\tau_{ij}(0)$ = 0,25. Для решения задачи нахождения маршрута, содержащего m вершин, и значения x, для которого $F(x)$ = min, разместим M муравьёв в случайно выбранные вершины графа, например в вершины 5, 10, 30, 39. Так как на t = 1 итерации веса рёбер одинаковы, определим случайным образом 4 маршрута.

$T_1(1)$: 5 — 12 — 18 — 25 $L_1(1) = 0{,}6 + 0{,}42 + 0{,}61 + 0{,}72 = 2{,}35$

$T_2(1)$: 10 — 22 — 18 — 36 $L_2(1) = 0{,}3 + 0{,}59 + 0{,}61 + 0{,}81 = 2{,}31$

$T_3(1)$: 30 — 25 — 17 — 40 $L_3(1) = 0{,}77 + 0{,}72 + 0{,}53 + 0{,}33 = 2{,}35$

$T_4(1)$: 39 — 37 — 11 — 24 $L_4(1) = 0{,}90 + 0{,}14 + 0{,}18 + 0{,}21 = 1{,}43$

Таким образом, на рёбрах маршрута k количество отложенного феромона после 1 итерации $\tau_{ij}^k(0) + \Delta\tau_{ij}^k = \tau_{ij}^k(0) + Q/L_k(1)$ составит:

$\tau_{ij}^1(0) + \Delta\tau_{ij}^1 = 0{,}25 + 1{,}7 = 1{,}95$; $\tau_{ij}^2(0) + \Delta\tau_{ij}^2 = 0{,}25 + 1{,}73 = 1{,}98$;

$\tau_{ij}^3(0) + \Delta\tau_{ij}^3 = 0{,}25 + 1{,}7 = 1{,}95$; $\tau_{ij}^4(0) + \Delta\tau_{ij}^4 = 0{,}25 + 2{,}8 = 3{,}05$.

После испарения получим следующие значения концентрации на рёбрах графа, соответствующих маршрутам: $\tau_{ij}^1(1) = 0{,}78$; $\tau_{ij}^2(1) = 0{,}79$; $\tau_{ij}^3(1) = 0{,}78$; $\tau_{ij}^4(1) = 1{,}22$. Рёбра графа после 1-й итерации будут иметь веса: $\tau_{5,12}(1) = \tau_{12,18}(1) = \tau_{18,25}(1) = \tau_{30,25}(1) = \tau_{25,17}(1) = \tau_{17,40}(1) =$ $= 0{,}78$; $\tau_{10,22}(1) = \tau_{22,18}(1) = \tau_{18,36}(1) = 0{,}79$; $\tau_{39,37}(1) = \tau_{37,11}(1) = \tau_{11,24}(1) = 1{,}22$. При этом $F_1 = \min\limits_{x_i \in T_1(1)}\left(F(x_i)\right) = F(12) = 0{,}42$; $F_2 = F(10) = 0{,}3$; $F_3 = F(40) = 0{,}33$; $F_4 = F(37) = 0{,}14$; $F^* =$ $= 0{,}14$. Для остальных рёбер веса $\tau_{ij}(1) = 0{,}25$.

Перейдём к следующей итерации при t = 2. Разместим муравьёв в случайно выбранные позиции, например, в 33, 42, 44, 10, и пусть определены следующие маршруты.

$T_1(2)$: 33 — 27 — 37 — 11 $L_1(2) = 0{,}06 + 0{,}07 + 0{,}14 + 0{,}18 = 0{,}45$

$T_2(2)$: 42 — 9 — 39 — 37 $L_2(2) = 0{,}26 + 0{,}22 + 0{,}90 + 0{,}14 = 1{,}52$

$T_3(2)$: 44 — 10 — 26 — 27 $L_3(2) = 0{,}30 + 0{,}30 + 0{,}35 + 0{,}07 = 1{,}02$

$T_4(2)$: 10 — 13 — 25 — 17 $L_4(2) = 0{,}30 + 0{,}69 + 0{,}72 + 0{,}53 = 2{,}24$

Таким образом, на рёбрах маршрутов количество отложенного феромона составит:

$\Delta\tau_{ij}^1 = Q/L_1(2) = 4/0{,}45 = 8{,}89$; $\Delta\tau_{ij}^2 = Q/L_2(2) = 4/1{,}52 = 2{,}63$;

$\Delta\tau_{ij}^3 = Q/L_3(2) = 4/1{,}02 = 3{,}92$; $\Delta\tau_{ij}^4 = Q/L_4(2) = 4/2{,}24 = 1{,}79$.

Таким образом, после испарения феромона рёбра графа будут иметь следующие веса: $\tau_{5,12}(2) = \tau_{12,18}(2) = \tau_{18,25}(2) = \tau_{30,25}(2) = \tau_{25,17}(2) = \tau_{17,40}(2) = 0{,}31$; $\tau_{10,22}(2) = \tau_{22,18}(2) =$ $= \tau_{18,36}(2) = 0{,}32$; $\tau_{10,13}(2) = \tau_{13,25}(2) = \tau_{25,17}(2) = 0{,}82$; $\tau_{44,10}(2) = \tau_{10,26}(2) = \tau_{26,27}(2) = 1{,}67$; $\tau_{42,9}(2) = \tau_{9,39}(2) = 1{,}15$; $\tau_{39,37}(2) = 1{,}54$; $\tau_{33,27}(2) = \tau_{27,37}(2) = 3{,}66$; $\tau_{37,11}(2) = 4{,}04$; $\tau_{11,24}(2) =$ $= 0{,}49$.

После 2-й итерации $F_1 = F(33) = 0{,}06$; $F_2 = F(9) = 0{,}22$; $F_3 = F(27) = 0{,}07$; $F_4 = F(10) =$ $= 0{,}3$, $F^* = 0{,}06$. Для остальных рёбер веса $\tau_{ij}(2) = 0{,}25$.

Таким образом, данная итерация показывает, как маршрут, проходящий по вершинам с наименьшим весом, максимально обогащается феромоном.

Перейдём к итерации t = 3 и разместим муравьёв в вершинах 19, 8, 32, 10. Пусть определены следующие маршруты.

$T_1(3)$: 19 — 37 — 27 — 33 $L_1(3) = 0{,}0 + 0{,}14 + 0{,}07 + 0{,}06 = 0{,}27$

$T_2(3)$: 8 — 13 — 10 — 22 $L_2(3) = 0{,}63 + 0{,}69 + 0{,}30 + 0{,}59 = 2{,}21$

$T_3(3)$: 32 — 33 — 27 — 37 $L_3(3) = 0{,}91 + 0{,}06 + 0{,}07 + 0{,}14 = 1{,}18$

$T_4(3)$: 10 — 17 — 39 — 37 $L_4(3) = 0{,}30 + 0{,}53 + 0{,}90 + 0{,}14 = 1{,}87$

Таким образом, после 3-й итерации количество отложенного феромона на рёбрах, входящих в маршруты, составит

$\Delta\tau_{ij}^{1} = Q/L_1(3) = 4/0,27 = 14,81;$ $\Delta\tau_{ij}^{2} = Q/L_2(3) = 4/2,21 = 1,80;$

$\Delta\tau_{ij}^{3} = Q/L_3(3) = 4/1,18 = 3,39;$ $\Delta\tau_{ij}^{4} = Q/L_4(3) = 4/1,87 = 2,14.$

Таким образом, после испарения феромона рёбра графа будут иметь следующие веса: $\tau_{5,12}(3) = \tau_{12,18}(3) = \tau_{18,25}(3) = \tau_{30,25}(3) = \tau_{25,17}(3) = \tau_{17,40}(3) = 0,124$; $\tau_{10,22}(3) = 0,84$; $\tau_{22,18}(3) = \tau_{18,36}(3) = 0,128$; $\tau_{39,37}(3) = 1,47$; $\tau_{37,11}(3) = 1,62$; $\tau_{11,24}(3) = 0,20$; $\tau_{42,9}(3) = \tau_{9,39}(3) = 0,46$; $\tau_{44,10}(3) = \tau_{10,26}(3) = \tau_{26,27}(3) = 0,67$; $\tau_{13,25}(3) = \tau_{25,17}(3) = 0,33$; $\tau_{19,37}(3) = 6,02$; $\tau_{37,27}(3) = \tau_{27,33}(3) = 8,74$; $\tau_{8,13}(3) = 0,82$; $\tau_{13,10}(3) = 1,05$; $\tau_{32,33}(3) = 1,46$; $\tau_{10,17}(3) = \tau_{17,39}(3) = 0,96$.

После 3-й итерации $F_1 = F(19) = 0$; $F_2 = F(10) = 0,30$; $F_3 = F(33) = 0,06$; $F_4 = F(37) = 0,14$; $F^* = 0$. Для остальных рёбер веса $\tau_{ij}(3) = 0,25$.

Таким образом, данный пример демонстрирует, как на 3-й итерации маршрут, содержащий вершины с наименьшим весом (19 — 37 — 27 — 33), наиболее сильно обогащается феромоном. Значения, соответствующие этим вершинам, являются, очевидным образом, делителями числа N с максимальной степенью точности ε (для данных вершин $0 \le \varepsilon \le 0,14$). Очевидно, что на последующих итерациях по этому маршруту пойдёт наибольшее количество агентов, в то время как другие пути будут исчезать.

Таким образом, отличительными особенностями данного подхода являются:

- возможность эффективной параллельной реализации, связанной с отсутствием миграции особей между процессорами (в отличие от классического ГА);

- возможность определения множества значений x_i, являющихся наилучшим приближением делителя числа N с заданной степенью точности.

Отметим, что, как следует из приведённого примера, веса вершин графа могут определяться в процессе работы алгоритма (при формировании маршрута муравьёв), т. е. после этого задача сводится к реализации классического муравьиного алгоритма для определения кратчайшего маршрута заданной длины.

Экспериментальные результаты показывают, что эффективность муравьиных алгоритмов растёт с увеличением размерности решаемых задач оптимизации. При этом сходимость и качество решения зависят от начального расположения колонии и выбранных параметров. В качестве теоретической оценки размерности колонии для данной задачи можно принять $M > (n_j - n_i + 1) / m$. Так как в этом случае маршруты муравьёв с высокой степенью вероятности пересекаются, также высока и вероятность того, что муравей повернёт на оптимальный путь, пройдёт по нему и обогатит его феромоном.

Алгоритм разложения составных чисел на простые сомножители с использованием пчелиных колоний. Необходимо решить следующую задачу. На отрезке $[n_i, n_j]$ определить целочисленные делители числа N, являющиеся простыми числами, т. е. осуществить разложение числа N на простые множители. Поскольку простые числа, как отмечено в [6, 9], на заданном интервале распределены по логарифмическому закону, то целью поиска является определение в r-окрестности точки x всех простых чисел y_i и точности ε, с которой данное простое число y_i является делителем числа N. Как и в предыдущем случае, значение ε определим как значение функции $F(y_i) = (N / y_i) - [N / y_i]$.

Рассмотрим применение для реализации данного подхода алгоритмов пчелиных колоний, являющихся относительно новым «природным» алгоритмом, используемым, как отмечено в [14], в первую очередь, для оптимизации сложных многомерных функций. Так как в данном случае ищется экстремум немонотонной функции $F(x)$, поэтому исследование возможности применения для решения данной задачи эвристических методов, не использующих непосредственным образом

аппарат математического анализа, является, несомненно, актуальной задачей. Отметим, что описание алгоритма, основанного на поведении колонии пчёл, приводится в [14, 15, 16]. Исследование пчелиных алгоритмов для решения комбинаторных теоретико-графовых задач (задача разбиения графа, раскраска графа, сравнение с другими биоинспирированными методами) приводится в [17, 18]. Таким образом, на основе математической модели алгоритма, основанного на поведении колонии пчёл, и его описания в [16] алгоритм факторизации числа сформулируем в следующей форме. Как и ранее, будем предполагать, что поиск простого делителя x_j осуществляется на заданном отрезке $[n_i, n_j]$.

1. Определить параметры алгоритма: количество пчёл-разведчиков D, количество рабочих пчёл B, количество участков для исследования окрестностей Z, точность нахождения делителя ε.

2. Выбрать на отрезке $[n_i, n_j]$ D значений аргумента $x_1, \ldots, x_D$.

3. В выбранные точки x_i направить B рабочих пчёл для поиска в их r-окрестности простых чисел в соответствии со следующим алгоритмом [6].

3.1. Определить для каждого значения x_i значение окрестности $r = n / 1{,}442695$ (где n — число бит в двоичной записи числа).

3.2. Каждое число $y \in [x_i - r, x_i + r]$ последовательно проверяется на делимость с простыми числами в интервале $[2, 2 \cdot r]$.

3.3. К числам, которые прошли тест проверки делимости, применяются известные тесты проверки простоты числа, описанные в [19], например, тест Миллера — Рабина, алгоритм, основанный на матрице Сандарама [20].

4. После определения множества простых чисел Y для каждого $y_i \in Y$ определить значение функции $F(y_i)$, найти $\min\limits_{y_i \in Y} F(y_i)$. Определить y_i, для которых $F(y_i) < \varepsilon$.

5. Из множества Y выбрать случайным образом Z элементов, данные значения обозначить как $x_1, \ldots, x_Z$. Отправить D пчёл-разведчиков для поиска на отрезке $[n_i, n_j]$ D значений аргумента $x_{Z+1}, \ldots, x_{Z+D}$. Если условия остановки не выполнены, переход к 3, иначе к 6.

6. Конец работы алгоритма.

Условиями остановки алгоритма могут являться окончание временного ресурса; определение величины (или множества величин) x_i, для которых $F(x_i) = 0$ или $F(x_i) < \varepsilon$; определение значений функции $F(x_i)$ для всех $x_i \in [n_i, n_j]$.

Таким образом, в данном алгоритме выбор значений аргумента $x_i \in [n_i, n_j]$ имитирует поведение пчёл-разведчиков, а поиск в r-окрестности наиболее вероятных простых чисел имитирует поведение рабочих пчёл (пчёл-фуражиров). Поскольку в данном случае определяется экстремум немонотонной функции, то выбор точек x_i на отрезке $[n_i, n_j]$ для поиска простых чисел в их окрестности производится на каждой итерации случайным образом, что приводит в общем случае к равновероятной возможности получения глобального оптимума на каждой итерации (в отличие от направленного схождения к экстремуму в классическом пчелином алгоритме, описанном в [16]).

Отличительной особенностью разработанного алгоритма, как и в [16], является динамическое разбиение поискового пространства на случайные области, что, с одной стороны, уменьшает время работы алгоритма, а с другой — повышает вероятность нахождения глобального оптимума на каждой итерации. В то же время благодаря случайному равновероятному поиску по всей длине отрезка становится возможным применение эффективных стратегий распараллеливания, приводящих к сокращению временных затрат. На каждой итерации временные затраты равны поиску в самой большой r-окрестности.

Рассмотрим пример применения итерационного алгоритма. Пусть $N = 15589$ и в памяти процессора содержится отрезок $[n_i, n_j] = [115, 140]$. Определим $D = 2$ и выберем случайным об-

разом точки $x_1 = 123$, $x_2 = 133$. Так как $x_1 = (123)_{10} = (1111011)_2$, то определим $n = 7$ и $r = n / 1{,}442695 = 5$.

Выделим интервал чисел [118, 128]. Проверим делимость чисел в данном интервале на простые числа в интервале $[2, 2 \cdot r] = [2, 10]$, т. е. на {2, 3, 5, 7}. Получим числа {121, 127}. Однако, т. к. число $(121 - 1) / 2 = 60$ содержится в матрице Сандарама в позиции (5, 5), то число 121 является составным. Для числа 127 $F(127) = 0{,}75$.

Так как $x_2 = (133)_{10} = (10000101)_2$, то $n = 8$ и $r = 6$.

Рассмотрим интервал чисел [127, 139]. Проверим делимость чисел из интервала на простые числа в интервале [2, 12], т. е. на {2, 3, 5, 7, 11}. Получим числа $Y = \{127, 131, 137, 139\}$. Так как для всех $y_i \in Y$ $(y_i - 1) / 2$ не содержится в матрице Сандарама, то данные числа являются простыми. При этом $F(127) = 0{,}75$; $F(131) = 0{,}0$; $F(137) = 0{,}79$; $F(139) = 0{,}15$. Таким образом, на данной итерации определено значение $x_i \in [n_i, n_j]$, являющееся точным делителем числа N.

Отметим, что в общем случае задачу проверки простоты числа N с помощью матрицы Сандарама можно свести к проверке принадлежности числа $N' = (N - 1) / 2$ множеству арифметических прогрессий, составляющих строки (столбцы) матрицы, приведённой в [20]. То есть задача сводится к определению целочисленного n из соотношений
$4 + 3 \cdot (n - 1) = N'$, $7 + 5 \cdot (n - 1) = N'$, …, $a_k + d_k \cdot (n - 1) = N'$, где $a_k = 4 + 3 \cdot (k - 1)$, $d_k = 3 + 2 \cdot (k - 1)$, т. е. $n - 1 = (N' - (4 + 3 \cdot (k - 1))) / (3 + 2 \cdot (k - 1))$.

Поскольку определение целого значения n имеет смысл при

$$\frac{N' - \left(4 + 3 \cdot \left(k - 1\right)\right)}{3 + 2 \cdot \left(k - 1\right)} > 1, \tag{5}$$

то определим значение k — число прогрессий (строк или столбцов матрицы), которые надо проверить для определения, является ли N' основой простого числа (в предположении, что $N' \neq 4$ не является первым членом первой прогрессии). Преобразуя (5), получим $N' - 3 \cdot k - 1 > 2 \cdot k + 1$, откуда

$$k < \left\rceil \frac{N' - 2}{5} \right\lceil, \tag{6}$$

где $\rceil x \lceil$ — ближайшее целое снизу к числу x. Таким образом, если число N' не является членом первых k арифметических прогрессий матрицы Сандарама (где k определяется выражением (6)), то число N достоверно является простым. Отметим, что для предлагаемого в [6, 19] вероятностного теста Миллера — Рабина рекомендуется в общем случае проверка $q = \log_2 N$ свидетелей простоты, после чего вероятность того, что N составное, не превышает 4^{-q}.

Поскольку наиболее целесообразными являются криптосистемы, в которых простые сомножители P и Q, составляющие модуль M, имеют порядок 2^{512}, то максимальное значение $r = 355$, поэтому длина интервала $|2, 2 \cdot r| = 700$. Простые числа в этом промежутке могут быть найдены с помощью известных таблиц, поэтому их определение не влияет на временную сложность алгоритма.

Выводы. Таким образом, в данной работе был представлен возможный подход к факторизации составных чисел с использованием биоинспирированных методов (алгоритмов муравьиных и пчелиных колоний), описаны основные отличительные особенности методов (в частности, возможность эффективной параллельной реализации), были также представлены демонстрационные примеры, иллюстрирующие возможность практического использования предложенных алгоритмов.

Библиографический список

1. Зайцев, А. А. Обзор эволюционных методов оптимизации на основе роевого интеллекта / А. А. Зайцев, В. В. Курейчик, А. А. Полупанов // Известия ЮФУ. — 2010. — № 12 (113). — С. 7—12.

2. Лебедев, О. Б. Трассировка в канале методом муравьиной колонии / О. Б. Лебедев // Известия ЮФУ. — 2009. — № 4 (93). — С. 46—52. (Интеллектуальные САПР).

3. Романец, Ю. В. Защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев, В. Ф. Шаньгин. — М.: Радио и связь, 2001. — 376 с.

4. Беляев, А. В. Методы и средства защиты информации / А. В. Беляев. Электрон. ресурс. Режим доступа: http://docs.luksian.com/security/articles/methods/ (дата обращения 27.12.2011).

5. Основные тенденции развития открытой криптографии. Электрон. ресурс. Режим доступа: http://bre.ru/security/12050.html (дата обращения 27.12.2011).

6. Кажаров, А. А. Разработка модели криптоанализа RSA при помощи генетических алгоритмов / А. А. Кажаров, Х. А. Кажаров. Электрон. ресурс. Режим доступа: http://www.contrterror.tsure.ru/index.php/index.php?option=com_content&view=article&id=13 (дата обращения 28.11.2011).

7. Сергеев, А. С. О возможности применения методов генетического поиска для реализации криптоанализа асимметричного алгоритма шифрования данных RSA / А. С. Сергеев // Изв. вузов. Сев.-Кавк. регион. Техн. науки. — 2008. — № 3. — С. 48—52.

8. Чернышёв, Ю. О. Исследование и разработка методов генетического поиска для реализации криптоанализа алгоритма IDEA и решения основных теоретико-числовых задач криптографии / Ю. О. Чернышёв, А. С. Сергеев, Н. Н. Венцов // Вестник РГУПС. — 2009. — № 3 (35). — С. 70—79.

9. Цагир, Д. Первые 50 миллионов простых чисел / Д. Цагир. Электрон. ресурс. Режим доступа: http://ega-math.narod.ru/Liv/Zagier.htm (дата обращения 25.11.2011).

10. Дискретная математика: алгоритмы. Электрон. ресурс. Режим доступа: http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005 (дата обращения 18.12.2011).

11. Кажаров, А. А. Муравьиные алгоритмы для решения транспортных задач / А. А. Кажаров, В. М. Курейчик // Известия РАН. Теория и системы управления. — 2010. — № 1. — С. 32—45.

12. Муравьиные алгоритмы. Электрон. ресурс. Режим доступа: http://rain.ifmo.ru/cat/data/theory/unsorted/ant-algo-2006/article.pdf (дата обращения 27.12.2011).

13. Алгоритмы муравьиной колонии. Электрон. ресурс. Режим доступа: http://www.wikiznanie.ru/ru-wz/index.php/Алгоритмы_муравьиной_колонии (дата обращения 27.12.2011).

14. Алгоритм пчёл для оптимизации функции. Электрон. ресурс. Режим доступа: http://jenyay.net/Programming/Bees (дата обращения: 27.12.2011).

15. Алгоритм пчёл для оптимизации функции. Электрон. ресурс. Режим доступа: http://lit999.narod.ru/soft/ga/index.html (дата обращения: 27.12.2011).

16. Курейчик, В. В. Роевой алгоритм в задачах оптимизации / В. В. Курейчик, Д. Ю. Запорожец // Известия ЮФУ. — 2010. — № 7 (108). — С. 28—32.

17. Курейчик, В. М. Использование пчелиных алгоритмов для решения комбинаторных задач / В. М. Курейчик, А. А. Кажаров. Электрон. ресурс. Режим доступа: http://www.nbuv.gov.ua/portal/natural/ii/2010_3/AI_2010_3/6/00_Kureychik_Kazharov.pdf (дата обращения: 27.12.2011).

18. Курейчик, В. М. Применение пчелиных алгоритмов для раскраски графов / В. М. Курейчик, А. А. Кажаров // Известия ЮФУ. — 2010. — № 12 (113). — С. 7—12.

19. Тест простоты / Википедия. Электрон. ресурс. Режим доступа: http://ru.wikipedia.org/wiki/Тест_простоты (дата обращения: 27.12.2011).

20. Аврутин, В. А. Алгоритм поиска простых чисел в заданном интервале / В. А. Аврутин. Электрон. ресурс. Режим доступа: http://library.mephi.ru/data/scientific-sessions/2003/12/024.html (дата обращения 17.11.2011).

Материал поступил в редакцию 27.12.2011.

References

1. Zajcev, A. A. Obzor e`volyucionny`x metodov optimizacii na osnove roevogo intellekta / A. A. Zajcev, V. V. Kurejchik, A. A. Polupanov // Izvestiya YuFU. — 2010. — # 12 (113). — S. 7—12. — In Russian.

2. Lebedev, O. B. Trassirovka v kanale metodom murav`inoj kolonii / O. B. Lebedev // Izvestiya YuFU. — 2009. — # 4 (93). — S. 46—52. (Intellektual`ny`e SAPR). — In Russian.

3. Romanecz, Yu. V. Zashhita informacii v komp`yuterny`x sistemax i setyax / Yu. V. Romanecz, P. A. Timofeev, V. F. Shan`gin. — M.: Radio i svyaz`, 2001. — 376 s. — In Russian.

4. Belyaev, A. V. Metody` i sredstva zashhity` informacii / A. V. Belyaev. E`lektron. resurs. Rezhim dostupa: http://docs.luksian.com/security/articles/methods/ (data obrashheniya: 27.12.2011). — In Russian.

5. Osnovny`e tendencii razvitiya otkry`toj kriptografii. E`lektron. resurs. Rezhim dostupa: http://bre.ru/security/12050.html (data obrashheniya: 27.12.2011). — In Russian.

6. Kazharov, A. A. Razrabotka modeli kriptoanaliza RSA pri pomoshhi geneticheskix algoritmov / A. A. Kazharov, X. A. Kazharov. E`lektron. resurs. Rezhim dostupa: http://www.contrterror.tsure.ru/index.php/index.php?option=com_content&view=article&id=13 (data obrashheniya: 28.11.2011). — In Russian.

7. Sergeev, A. S. O vozmozhnosti primeneniya metodov geneticheskogo poiska dlya realizacii kriptoanaliza asimmetrichnogo algoritma shifrovaniya danny`x RSA / A. S. Sergeev // Izv. vuzov. Sev.-Kavk. region. Texn. nauki. — 2008. — # 3. — S. 48—52. — In Russian.

8. Cherny`shyov, Yu. O. Issledovanie i razrabotka metodov geneticheskogo poiska dlya realizacii kriptoanaliza algoritma IDEA i resheniya osnovny`x teoretiko-chislovy`x zadach kriptografii / Yu. O. Cherny`shyov, A. S. Sergeev, N. N. Venczov // Vestnik RGUPS. — 2009. — # 3 (35). — S. 70—79. — In Russian.

9. Czagir, D. Pervy`e 50 millionov prosty`x chisel / D. Czagir. E`lektron. resurs. Rezhim dostupa: http://ega-math.narod.ru/Liv/Zagier.htm (data obrashheniya: 25.11.2011). — In Russian.

10. Diskretnaya matematika: algoritmy`. E`lektron. resurs. Rezhim dostupa: http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005 (data obrashheniya: 18.12.2011). — In Russian.

11. Kazharov, A. A. Murav`iny`e algoritmy` dlya resheniya transportny`x zadach / A. A. Kazharov, V. M. Kurejchik // Izvestiya RAN. Teoriya i sistemy` upravleniya. — 2010. — # 1. — S. 32—45. — In Russian.

12. Murav`iny`e algoritmy`. E`lektron. resurs. Rezhim dostupa: http://rain.ifmo.ru/cat/data/theory/unsorted/ant-algo-2006/article.pdf (data obrashheniya: 27.12.2011). — In Russian.

13. Algoritmy` murav`inoj kolonii. E`lektron. resurs. Rezhim dostupa: http://www.wikiznanie.ru/ru-wz/index.php/Алгоритмы_муравьиной_колонии (data obrashheniya: 27.12.2011). — In Russian.

14. Algoritm pchyol dlya optimizacii funkcii. E`lektron. resurs. Rezhim dostupa: http://jenyay.net/Programming/Bees (data obrashheniya: 27.12.2011). — In Russian.

15. Algoritm pchyol dlya optimizacii funkcii. E`lektron. resurs. Rezhim dostupa: http://lit999.narod.ru/soft/ga/index.html (data obrashheniya: 27.12.2011). — In Russian.

16. Kurejchik, V. V. Roevoj algoritm v zadachax optimizacii / V. V. Kurejchik, D. Yu. Zaporozhecz // Izvestiya YuFU. — 2010. — # 7 (108). — S. 28—32. — In Russian.

17. Kurejchik, V. M. Ispol`zovanie pcheliny`x algoritmov dlya resheniya kombinatorny`x zadach / V. M. Kurejchik, A. A. Kazharov. E`lektron. resurs. Rezhim dostupa: http://www.nbuv.gov.ua/portal/natural/ii/2010_3/AI_2010_3/6/00_Kureychik_Kazharov.pdf (data obrashheniya 24.01.2012). — In Russian.

18. Kurejchik, V. M. Primenenie pcheliny`x algoritmov dlya raskraski grafov / V. M. Kurejchik, A. A. Kazharov // Izvestiya YuFU. — 2010. — # 12 (113). — S. 7—12. — In Russian.

19. Test prostoty` / Vikipediya. E`lektron. resurs. Rezhim dostupa: http://ru.wikipedia.org/wiki/Тест_простоты (data obrashheniya 17.01.2012). — In Russian.

20. Avrutin, V. A. Algoritm poiska prosty`x chisel v zadannom intervale / V. A. Avrutin. E`lektron. resurs. Rezhim dostupa: http://library.mephi.ru/data/scientific-sessions/2003/12/024.html (data obrashheniya 17.11.2011). — In Russian.

CRYPTANALYSIS BIOINSPIRED METHODS OF ASYMMETRIC KEY ON THE BASIS OF COMPOSITE NUMBER FACTORIZATION

A. S. Sergeyev
(Don State Technical University),
O. P. Tretyakov, A. E. Vasilyev
(Krasnodar branch of Military Academy of Communication),
Y. O. Chernyshev
(Don State Technical University)

The application of the bioinspired methods for handling the cryptanalysis problem of the asymmetric encryption algorithms on the basis of the composite number factorization is considered. The algorithms of ant and bee colonies for the composite number factorization by the definition of the integer divisor to the specified accuracy in the stated interval are adduced. The properties of the methods presented, including the efficient parallel feasibility, are described.

Keywords: *cryptanalysis, bee algorithm, ant colony algorithm, pheromone, factorization of numbers, asymmetric cryptosystems, bioinspired methods.*