ПРОБЛЕМА ОЦЕНКИ ЭФФЕКТИВНОСТИ ТЕСТА МИЛЛЕРА-РАБИНА

Научная статья

Жуманиёзов А.Р.[1, *]
[1] ORCID : 0000-0002-6770-2184;
[1] Казанский федеральный университет, Казань, Российская Федерация

* Корреспондирующий автор (aszhumaniezov[at]kpfu.ru)

Аннотация

В данной работе проведён анализ эффективности теста Миллера-Рабина. В качестве отправной точки для анализа был выбран алгоритм поиска последовательности чисел ψ_n. В рамках данного алгоритма было обнаружено «бутылочное горлышко» и представлен способ его решения. Им оказался расход памяти.

Основной идеей выступает распределение простых чисел по значениям $bin(ord_p(a_i))$. Сделан вывод о неравномерности распределения и сильном смещении к меньшим значениям $bin(ord_p(a_i))$. Для наглядности все рассуждения и результаты экспериментов сопровождаются графиками.

Благодаря полученным данным был сделан вывод о возможном разбиении всего множества простых чисел на подмножества по значению $bin(ord_p(a_i))$. Таким образом, получен итоговый алгоритм, в котором оптимизирован расход памяти по сравнению с исходным.

Ключевые слова: тест Миллера-Рабина, строго псевдопростые числа, теория вероятностей.

THE PROBLEM OF EVALUATING THE EFFECTIVENESS OF THE MILLER-RABIN PRIMALITY TEST

Research article

Zhumaniezov A.R.[1, *]
[1] ORCID : 0000-0002-6770-2184;
[1] Kazan Federal University, Kazan, Russian Federation

* Corresponding author (aszhumaniezov[at]kpfu.ru)

Abstract

This work analyses the efficiency of the Miller-Rabin test. As a starting point for the analysis, the algorithm for finding a sequence of ψn numbers was chosen. Within this algorithm, a "bottle-neck" was detected and a way to solve it was presented. It turned out to be the memory consumption.

The main idea is the distribution of pseudoprime numbers over $bin(ordp(ai))$ values. It is concluded that the distribution is uneven and there is a strong bias towards smaller values of $bin(ordp(ai))$. For clarity, all reasoning and experimental results are accompanied by graphs.

Thanks to the data obtained, it was concluded that it is possible to divide the whole set of pseudoprime numbers into subsets by the value of $bin(ordp(ai))$. Thus, the final algorithm is produced, in which the memory consumption is optimized compared to the original algorithm.

Keywords: Miller-Rabin primality test, pseudoprime numbers, probability theory.

Введение

Современная криптография, а особенно её защищённость, основывается на различных свойствах простых чисел. Поэтому возникает необходимость в эффективном поиске достаточно больших простых чисел. Существует множество различных подходов для решения данной проблемы. Однако наиболее известным и эффективным является использование теста Миллера-Рабина.

Таким образом, актуальность работы обеспечивается использованием простых чисел в современных исследованиях. Например, в статьях [1] и [2] представлен новый протокол маршрутизации, основанный на простых числах, позволяющий обнаруживать кротовые норы в мобильных сетях. Также в статьях [3], [4] и [5] представлен алгоритм шифрования, основанный на простых числах и биометрии, а также его применение в технологии блокчейна, а также его использование в интернете вещей. А в статье [10] представлен алгоритм шифрования изображений, использующий множество простых чисел и полярное разложение.

Основной целью является исследование эффективности работы теста Миллера-Рабина и распределения его ошибки. Для достижения поставленной цели были сформулированы и решены следующие задачи:

1. Анализ существующих методов оценки эффективности теста Миллера-Рабина.
2. Анализ возможных оптимизаций для существующего метода оценки эффективности теста Миллера-Рабина.

Тест Миллера-Рабина [7], [8] является вероятностным тестом. Это означает, что тест может выносить ошибочный вердикт, но с очень маленькой вероятностью. В настоящий момент известна только верхняя граница для её значения, однако она сильно завышена.

Есть и другой подход к оценке эффективности алгоритма – последовательность чисел ψ_n – наименьшее строго псевдопростое число для n первых простых чисел. Главная сложность этого подхода – быстрый рост значение и отсутствие эффективных неперебороных алгоритмов.

Таким образом, имея достаточно точную информацию об эффективности теста Миллера-Рабина, можно на его основе создавать модификации и получать достаточно точные асимптотики времени выполнения и памяти.

К примеру, К. Нари, Е. Оздемир и Н. А. Озкирисци представили алгоритм [9], добавляющий дополнительные проверки после запуска теста Миллера-Рабина с основанием 2. Также Д. Соренсон и Д. Вебстер разработали алгоритм по поиску наборов простых чисел по заданному паттерну [10]. Для построения оценки эффективности они используют знания о распределении строго псевдопростых чисел.

Методы и принципы исследования

Тест Миллера-Рабина – вероятностный тест, представленный сперва Г. Миллером в 1976 г. [7], затем улучшенным М. О. Рабиным в 1980 [8]. Основан этот тест на модификации теоремы Эйлера [11].

Для упрощения дальнейшего описания теорем и алгоритмов введём следующие функции:

Определение 1. Пусть *n* – произвольное натуральное число, представимое следующий вид:

$$n = 2^5 * d, \text{ где } d - \text{нечётное число} \tag{1}$$

Тогда функции *bin(n)* и *odd(n)* определяются следующим образом:

$$\begin{aligned} bin(n) &= s \\ odd(n) &= d \end{aligned} \tag{2}$$

Тогда каждая итерация теста Миллера-Рабина заключается в выборе произвольного основания и проверки выполнимости следующих условий:

$$\begin{aligned} a^{odd(n-1)} &\equiv 1 (mod\ n) \\ \exists (0 \le i < bin(n-1)) | (a^{odd(n-1)})^{2^i} &\equiv -1 (mod\ n) \end{aligned} \tag{3}$$

Если одно из условий выполнилось, то число *a* называется свидетелем простоты числа *n*, а само число считается прошедшим текущую итерацию теста.

Первый подход к оценке эффективности алгоритма основан на вычислении количества свидетелей простоты произвольного числа. Все вычисления, производимые в рамках данного подхода основаны на следующей теореме [12]:

Теорема 1. Пусть *n* – произвольное натуральное число, представимое следующий вид:

$$n = u * v, \text{ где } (u, v) = 1 \tag{4}$$

Тогда выполняются все перечисленные условия:

$$\begin{aligned} ord_u(a) | \text{НОД}(\varphi(u), (u - \varphi(u))v - 1) \\ ord_v(a) | \text{НОД}(\varphi(v), (v - \varphi(v))u - 1) \\ bin(ord_u(a)) = bin(ord_v(a)) \end{aligned} \tag{5}$$

Где за $ord_k(a)$ обозначают порядок числа *a* по модулю *k*.

Обозначим за *W(n)* – количество свидетелей простоты. Ш.Т. Ишмухаметов, Б.Г. Мубараков и Р.Г. Рубцова представили конечную формулу [12] для расчёта функции *W(n)* для случая полупростого *n=p*q*:

$$W(n) = odd(d)^2 \ \frac{4^{bin(d)}+2}{3}, \text{ где } d = \text{НОД}(p-1,\ q-1) \tag{6}$$

Однако позже Б. Г. Мубараков получил формулу [13] для произвольного числа *n* по его разложению на простые множители $n = p_1^{r_1} * p_1^{r_1} * ... * p_k^{r_k}$:

$$\begin{aligned} d_i &= \text{НОД}(p_i - 1,\ \frac{n}{p_1^{r_i}} - 1) \\ s &= min(bin(d_i)) \\ W(n) &= \prod_{i=1}^{k} (odd(d_i)) * (1 + \sum_{j=o}^{s-1} 2^{kj}) \end{aligned} \tag{7}$$

Следующим шагом для оценки эффективности вводится функция *Fr(n)* – вероятность выбора свидетеля простоты. Поскольку из условий (3) следует, что *НОД (a, n)=1*, то значение функции будет вычисляться по следующей формуле:

$$Fr(n) = \frac{W(n)}{\varphi(n)} \tag{8}$$

М.О. Рабин доказал, что ¼ – верхняя граница [8] для *Fr(n)*. Однако данное значение достигается для бесконечного количества чисел *n*, что значительно усложняет анализ этой функции.

Поэтому оценки вычислялись для среднего значения вероятности на отрезке *Avg(Fr(n))*. Первая оценка [14] для этой функции была получена Б.Г. Мубараковым, но ограничиваясь только полупростыми числами *n=p*q* при фиксированном *p*:

$$Avg(Fr(n)) \le \frac{p^2}{2X} ln(ln(X)) ln(X) \tag{9}$$

Однако данная оценка была сильно завышена, поэтому Б.Г. Мубараков представил улучшение оценки, но для случая *p=2*p'*, где *p'* – простое число. Для этой цели рассматривались два возможных случая отношений *p* и *q*:

1. $q=(p-1)k+1$ → в этом случае верхняя оценка из (5) становится асимптотикой функции [14].
2. $q=2k+1$, где $2k \bmod (p-1) \neq 0$ → в этом случае верхняя оценка из (5) улучшается до следующей [14]:

$$Avg(Fr(n)) \ \frac{2}{X} ln(ln(X)) ln(X) \tag{10}$$

Наконец, посчитав математическое ожидание от обоих вариантов получаем результат [14]:

$$E(Avg(Fr(n))) \sim \frac{2p}{X} ln(ln(X)) ln(X) \quad (11)$$

После чего было высказано предположение, что для всех значений p оценка будет принимать следующий вид [14]:

$$Avg(Fr(n)) \sim C_p \frac{ln(ln(X)) ln(X)}{X} \quad (12)$$

Где коэффициент C_p находится на отрезке $[2, \frac{p^2}{2}]$.

Также были получены результаты [15] для трёхпростых чисел $n = p^* q^* r$. Первая оценка была получена для фиксированных p и q :

$$Avg(Fr(n)) \leq \frac{(pq)^2}{8X} \ln(\ln(X)) \ln(X) \quad (13)$$

Однако эта оценка также является сильно завышенной и требует улучшения. Одним из возможных способов разбиение всех чисел на группы, расчёт оценки для каждой группы и расчёт общего значения через математическое ожидание.

На настоящий момент получены оценки для двух классов:

1. Трёхпростое число, при фиксированном p и q , удовлетворяющих следующим условиям:

$$\begin{gathered} p-1=2p', \text{ где } p'\text{-простое число} \\ q-1=2q', \text{ где } q'- \text{ простое число} \\ \text{bin}(pq-1)=2 \\ rq \equiv 1(\text{mod} p') \\ rp \equiv 1(\text{mod} q') \\ r \equiv 1(\text{mod} pq-1) \end{gathered} \quad (14)$$

Итоговая оценка для такой ситуации [15], [16]:

$$\text{Avg}(Fr(n)) \sim \frac{pq^2}{8X} \ln(\ln(X)) \ln(X) \quad (15)$$

2. Трёхпростое число, при фиксированном p и q, удовлетворяющих следующим условиям:

$$\begin{gathered} p-1=2p', \text{ где } p'\text{-простое число} \\ q-1=2q', \text{ где } q'- \text{ простое число} \\ \text{bin}(pq-1)=2 \\ r \equiv 1 \ (\text{mod} p'^* q'^* (pq-1)) \end{gathered} \quad (16)$$

Итоговая оценка для такой ситуации [16]:

$$\text{Avg}(Fr(n)) \sim \frac{pq^2}{8X} \ln(\ln(X)) \ln(X) \quad (17)$$

Второй подход к оценке эффективности основан на свойствах функции $h(a,n)$ [17] определённой следующим образом:

$$h(a,n) = \begin{cases} a^u - 1, \text{ если } \text{bin}\left(\text{ord}_n(a)\right) = 0 \\ a^{u2^{c-1}}, \text{ если bin } \left(\text{ord}_n(a)\right) = c \end{cases} \quad (18)$$

Теперь укажем теорему, на которой построена другая оценка:

Теорема 2. Пусть n произвольное число, представимое в виде $n = k * p$, где p – простое. Тогда для того, чтобы n было строго псевдопростым по базе a необходима делимость $h(a,k)$ на p .

Используя теорему 2 и оценки количества делителей, мы получаем верхнюю оценку [18] для ошибки теста Миллера-Рабина, но только для полупростых чисел:

$$P(n = pq - \text{ строго псевдопростое по основанию } a) \leq \frac{\ln(2) \ln(a)}{(\ln(X))^2} \quad (19)$$

Однако реальные значения не достигают верхней оценки. Значит эту оценку можно улучшить.

Третий подход к оценке эффективности – вычисление последовательности чисел ψ_n – наименьшее строго псевдопростое число для n первых простых чисел. Ж. Женхианг смог получить кандидатов [19] для значений первых 19 элементов последовательности. До настоящего времени смогли доказать [20] эту гипотезу только для первых 13 элементов последовательности.

Рассматривая уже полученные результаты, можно сделать вывод о том, что последовательность очень быстро возрастает. Однако эффективный алгоритм поиска очередного алгоритма пока не найден, поэтому поиск следующего элемента последовательности затруднителен.

Наиболее оптимальный переборный алгоритм [20], который использовался для поиска последних известных элементов последовательности основан на Теореме 2 и следующих утверждениях:

Утверждение 1. Если для произвольных простых чисел p и q выполняется $\text{bin}(p-1) = \text{bin}(q-1)$ и $\text{bin}\left(ord_p(a)\right) = \text{bin}\left(ord_q(a)\right)$, то $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right)$.

Утверждение 2. Если для произвольных простых чисел p и q выполняется $\text{bin}(p-1) < \text{bin}(q-1)$ и $\text{bin}\left(\text{ord}_p(a)\right) = \text{bin}\left(ord_q(a)\right)$, то $\left(\frac{a}{q}\right) = 1$.

Сам алгоритм состоит из двух методов, каждый из которых выполняет перебор возможных кандидатов $n = pk$ по своим критериям.

Первый метод использует НОД для получения всех кандидатов. На вход метод получает число k. В ходе алгоритма вычисляются все $h(a_i, k)$ для всех оснований a_i. Затем вычисляется НОД полученных значений. p получаются путём перебора всех делителей НОД. Второй метод используя утверждения 1 и 2 перебирает все возможные остатки по достаточно большому модулю. После чего получаются перебором всех простых чисел с заданным остатком по модулю.

Поскольку первый метод эффективней на маленьких числах, а второй метод на больших, то первый метод применяется для всех $k \le \sqrt[3]{B}$, где B – верхняя граница отрезка на котором производится поиск, а второй метод для остальных k. Значение границы для методов было получено теоретическим путём для наиболее оптимального раздела.

Также для быстрого перебора k используется хэш-таблица в которой хранятся все простые числа p и значения $bin(ord_p(a_i))$ для всех оснований a_i. В качестве ключа используется хеш-значения от значений $bin(ord_p(a_i))$. Поскольку все значения из хэш-таблицы используются для формирования составных чисел, состоящих не менее чем из 3 делителей, то и хранить достаточно лишь все простые числа $p \le \sqrt[3]{B}$.

Таким образом итоговый алгоритм получается следующий:

Algorithm 1.

T – хэш-таблица, B – граница для перебора, a_i – набор оснований.

Перебираем все простые числа p от $max(a_i)+1$ до $\sqrt{B}$:

Вычисляем все $bin(ord_p(a_i))$.

Получаем список всех простых чисел из T с совпадающими значениями $bin(ord_p(a_i))$.

Формируем все возможные значения k:

Если $k \le \sqrt[3]{B}$, то перебираем всех кандидатов на строго псевдопростоту с помощью первого метода.

Иначе, с помощью второго.

Если $k \le \sqrt[3]{B}$, то добавляем в T вместе со всеми значениями $bin(ord_p(a_i))$.

Основные результаты

Поскольку последовательность чисел ψ_n растёт достаточно быстро, то рассматриваемые границы для поиска также будут сильно увеличиваться. Это приведёт не только к замедлению времени работы, но и увеличению объёма используемой памяти. Например, $\psi_{14} > 10^{27}$, следовательно, в хэш-таблице будет храниться порядка 10^9 элементов, что крайне много для хранения на компьютере. Поэтому необходимо модифицировать алгоритм 1, чтобы он затрагивал меньше памяти.

Для начала рассмотрим распределение простых чисел на отрезке по значению $bin(ord_p(a_i))$. Для ускорения вычисления мы введём следующую функцию:

Определение 1. Пусть p и a – произвольные числа, p – простое, а s определяется следующим образом:

$$a^{odd(p-1)*2^s} \equiv 1(mod\ p), \text{ но } a^{dd(p-1)*2^c} \not\equiv 1(mod\ p), \text{ для любого } 0 \le c < s \tag{20}$$

Тогда функция $bin_q(a,p)$, определяется следующим образом:

$$bin_1(a,p) = s \tag{21}$$

Эффективный поиск значения этой функции выполняется следующим алгоритмом:

Алгоритм 2.

p – простое, a – натуральное числа, $s = 0$

Посчитаем $p' = odd(p-a)$

Посчитаем $b_1 = a^{p'} mod\ p$

Пока b_1 не станет равным 1 выполняем:

$s = s + 1$

$b_1 = (b_1)^2\ mod\ p$

Значение s устанавливается результатом функции

Вычислительная сложность данного алгоритма $\Theta(log(p))$.

Для использования данного алгоритма докажем следующую теорему:

Теорема 3. Пусть n произвольное число, тогда выполняется следующее соотношение:

$$bin_1(a,p) \ne bin(ord_p(a)) \tag{22}$$

Доказательство. Пусть $bin_1(a,p) \ne bin(ord_p(a))$. Тогда возможно 2 варианта:

1. $bin_1(a,p) < bin(ord_p(a))$.

$(p-1)\vdots odd(ord_p(a)) \rightarrow НОД(ord_p(a), p-1) =$

$= НОД(2^{bin(ord_p(a))} odd(ord_p(a)), 2^{bin_1(p)} odd(p-1)) = 2^{bin_1(p)} odd(ord_p(a)) < ord_p(a)$

$a^{НОДord_p(a),p-1)} = a^{2^{bin_1(p)} odd(ord_p(a))} \equiv 1 mod\ p$ → $ord_p(a)$ найден неверно.

2. $bin_1(a,p) > bin(ord_p(a))$.

$(p-1)\vdots odd(ord_p(a)) \rightarrow odd(p-1)\vdots odd(ord_p(a)) \rightarrow u = \frac{odd(p-1)}{odd(ord_1(a))}$ – целое

$a^{2^{bin(ord_p(a))} * odd(p-1)} = a^{2^{bin(ord_p(a))} * odd(ord_p(a)) * u} = (a^{2^{bin(ord_p(a))} * odd(ord_p(a))})^u \equiv 1^u\ mod\ p = 1$

→ $bin_1(a,p)$ найден неверно.

Значит остаётся единственный вариант в (21).

Таким образом, заменив вычисление $bin(ord_p(a))$ на $bin_1(a,p)$, мы получим тот же результат, но за меньшее время.

Обсуждение

После получения распределения простых чисел на отрезке $[1, 10^9]$ по значениям $bin(ord_p(a))$ видно, что основная часть простых чисел имеет очень маленькое значение (см. рисунки 1, 2).

Рисунок 1 - Распределение простых чисел по значениям для основания 7

Рисунок 2 - Распределение простых чисел по значениям для основания 19

Также можно заметить, что количество простых чисел падает примерно в 2 раза, пока не приблизится к крайне малым значениям (см. рисунок. 3).

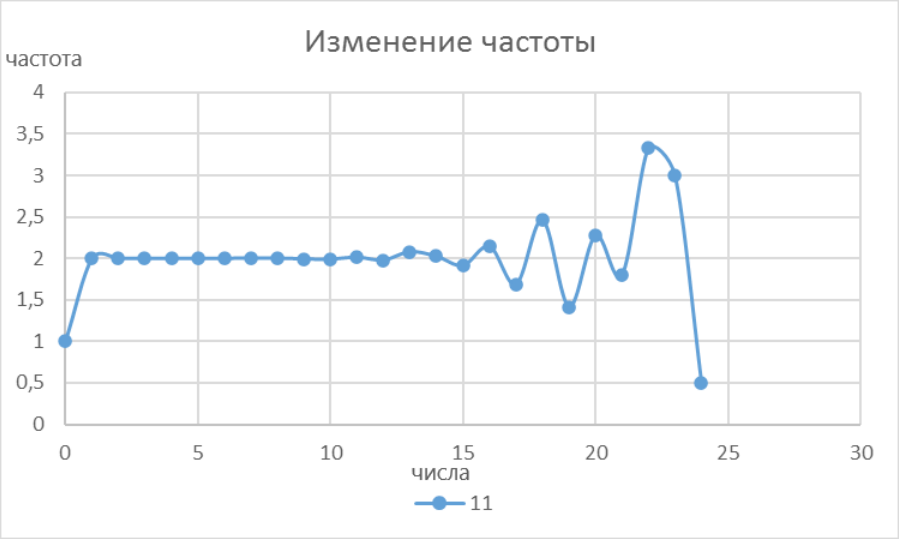

Рисунок 3 - Отношение количеств простых чисел на соседних значениях

Из-за столь большой скорости падения количества более 90% простых чисел имеет значение $bin(ord_p(a))$ не более 4 (см. рисунок. 4).

Рисунок 4 - Процент оставшегося количества чисел

Также можно заметить, что при оценке распределения $bin(ord_p(a))$ по нескольким значениям p оказывается, что при меньших границах количество увеличивается, а при больших уменьшается (см. рисунок. 5).

Рисунок 5 - Процент оставшегося количества чисел

Таким образом, можно сделать вывод о необходимости разбиения алгоритма на 2 этапа:

1 этап – перебрать все простые числа со значением всех $bin(ord_p(a))$ меньше заданного порогового значения X с помощью алгоритма 2. При этом само значение X перебирать из некоторого фиксированного набора $\{X_1, X_2, ..., X_k\}$.

2 этап – перебрать все простые числа со значением хотя бы одного $bin(ord_p(a))$ не меньше X с помощью другого, возможно менее эффективного для большого количества чисел, алгоритма.

Пример такого алгоритма:

Алгоритм 3.

P *– список простых чисел,* L *– список контейнеров для простых чисел*

Перебираем все основания a_i *:*

Очищаем все контейнеры в L

Перебираем все числа p *из* P *:*

Вычисляем $s = bin_1(a_i, p)$

Переносим p *из* P *в* $L[s]$

Переносим последовательно все простые числа из контейнеров L *в* P

Вычислительная сложность данного алгоритма $\Theta(|P|)$

Заключение

В рамках данной работы были представлены текущие результаты по оценке эффективности теста Миллера-Рабина. Было продемонстрировано три разных подхода, сделаны выводы о возможном направлении для каждого подхода. Для модификации был выбран третий подход – поиск последовательности чисел ψ_n – наименьшее строго псевдопростое число для n первых простых чисел.

Было обнаружено «бутылочное горлышко» для исходного алгоритма – им оказался расход памяти на хэш-таблицу. Поэтому дальнейшие исследования направлены на оптимизацию расхода памяти.

Были сделаны выводы о распределении простых чисел по значениям $bin(ord_p(a_i))$ и рассмотрен вариант использования полученных данных для модификации алгоритма.

Была представлена новая схема алгоритма со сравнимым временем работы, но уменьшенным расходом памяти.

Финансирование

Работа выполнена за счет средств Программы стратегического академического лидерства Казанского (Приволжского) федерального университета («ПРИОРИТЕТ-2030»).

Funding

The work was supported by the Strategic Academic Leadership Program of the Kazan (Volga Region) Federal University ("PRIORITET-2030").

Конфликт интересов

Не указан.

Conflict of Interest

None declared.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы / References

1. Muruganandam S. A Survey: Comparative Study of Security Methods and Trust Manage Solutions in MANET . / S. Muruganandam, J. Renjit, R. Kumar // Proceedings of the 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM); — Chennai: IEEE, 2019. — p. 125-131.
2. Sharma S. EPPN: Extended Prime Product Number based wormhole DETECTION scheme for MANETs. / S. Sharma, R.M. Sharma // Proceedings of the 2017 11th International Conference on Intelligent Systems and Control (ISCO); — Coimbatore: IEEE, 2017. — p. 251-254.
3. Iovane G. A Novel Blockchain Scheme Combining Prime Numbers and Iris for Encrypting Coding. / G. Iovane, M. Nappi, M. Chinnici et al. // Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech); — Fukuoka: IEEE, 2019. — p. 609-618.
4. Iovane G. Blockchain-Based Iris Authentication in Order to Secure IoT Access and Digital Money Spending. / G. Iovane, A. Rapuano, P. Di Gironimo // Pattern Recognition. ICPR International Workshops and Challenges. ICPR 2021; — Issue 12665. — Berlin: Springer, 2021. — p. 427-441.
5. Maddalena L. Pattern Recognition and beyond: Alfredo Petrosino’s Scientific Results. / L. Maddalena, M. Gori, S.K. Pal // Pattern Recognition Letters. — 2020. — 138. — p. 659–669.
6. Singh S.P. A Secure Image Encryption Algorithm Based on Polar Decomposition. / S.P. Singh, G. Bhatnagar, D.K. Gurjar // Proceedings of the 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA); — Penang: IEEE, 2018. — p. 135-139.
7. Miller G. Riemann’s Hypothesis and Tests for Primality. / G. Miller // Journal of Computer and System Sciences. — 1976. — 13. — p. 300-317.
8. Rabin M.O. Probabilistic Algorithm for Testing Primality. / M.O. Rabin // Journal of Number Theory. — 1980. — 12. — p. 128-138.
9. Nari K. Strong Pseudo Primes to Base 2 [Electronic source] / K. Nari, E. Ozdemir, N.A. Ozkirisci // arXiv. — 2019. — URL: https://arxiv.org/abs/1905.06447. (accessed: 25.03.23)
10. Sorenson P.J. Two Algorithms to Find Primes in Patterns [Electronic source] / P.J. Sorenson, J. Webster // arXiv. — 2019. — URL: https://arxiv.org/abs/1807.08777. (accessed: 25.03.23)
11. Ribenboim P. The New Book of Prime Number Records / P. Ribenboim — New York: Springer, 1995. — 541 p.
12. Ishmukhametov S.T. On the Number of Witnesses in the Miller–Rabin Primality Test. / S.T. Ishmukhametov, B.G. Mubarakov // Symmetry. — 2020. — 12.
13. Мубараков Б.Г. Эффективная оценка теста простоты Миллера-Рабина натуральных чисел / Б.Г. Мубараков // Материалы XIX Всероссийской молодежной научной школы-конференции; — Вып. 59. — Казань, 2020. — с. 106-109.
14. Mubarakov B.G. On the Number of Primality Witnesses of Composite Integers. / B.G. Mubarakov // Russian Mathematics. — 2021. — 65. — p. 73-77.
15. Zhumaniezov A.R. The Problem of Error Frequency Distribution in the Miller-Rabin Test For Tripleprime Numbers. / A.R. Zhumaniezov // Proceedings of the 10th International Conference on Foundations of Computer Science & Technology; — Issue 12. — Zurich: AIRCC Publishing Corporation, 2022.
16. Zhumaniezov A.R. Estimating the Distribution of Witnesses of the Primality of the Miller-Rabin Test. / A.R. Zhumaniezov // International Journal on Computational Science & Applications. — 2022. — 12.
17. Bleichenbacher D. Efficiency and Security of Cryptosystems Based on Number Theory dis....of PhD in Engineering: - : defense of the thesis 1996-01-01 : approved 1996-01-01 / D. Bleichenbacher — Zurich: 1996. — 98 p.
18. Жуманиёзов А.Р. Оценка вероятности ошибки теста Миллера-Рабина на полупростых числах. / А.Р. Жуманиёзов // Материалы Международного молодежного научного форума «ЛОМОНОСОВ-2022»; — Moscow: MAKS Press, 2022.
19. Zhang Z. Two Kinds of Strong Pseudoprimes up to 10^36. / Z. Zhang // Mathematics of Computation. — 2007. — 76. — p. 2095-2107.
20. Sorenson J. Strong Pseudoprimes to Twelve Prime Bases. / J. Sorenson, J. Webster // Mathematics of Computation. — 2015. — 86. — p. 985-1003.

Список литературы на английском языке / References in English

1. Muruganandam S. A Survey: Comparative Study of Security Methods and Trust Manage Solutions in MANET . / S. Muruganandam, J. Renjit, R. Kumar // Proceedings of the 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM); — Chennai: IEEE, 2019. — p. 125-131.
2. Sharma S. EPPN: Extended Prime Product Number based wormhole DETECTION scheme for MANETs. / S. Sharma, R.M. Sharma // Proceedings of the 2017 11th International Conference on Intelligent Systems and Control (ISCO); — Coimbatore: IEEE, 2017. — p. 251-254.
3. Iovane G. A Novel Blockchain Scheme Combining Prime Numbers and Iris for Encrypting Coding. / G. Iovane, M. Nappi, M. Chinnici et al. // Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech); — Fukuoka: IEEE, 2019. — p. 609-618.
4. Iovane G. Blockchain-Based Iris Authentication in Order to Secure IoT Access and Digital Money Spending. / G. Iovane, A. Rapuano, P. Di Gironimo // Pattern Recognition. ICPR International Workshops and Challenges. ICPR 2021; — Issue 12665. — Berlin: Springer, 2021. — p. 427-441.
5. Maddalena L. Pattern Recognition and beyond: Alfredo Petrosino’s Scientific Results. / L. Maddalena, M. Gori, S.K. Pal // Pattern Recognition Letters. — 2020. — 138. — p. 659–669.

6. Singh S.P. A Secure Image Encryption Algorithm Based on Polar Decomposition. / S.P. Singh, G. Bhatnagar, D.K. Gurjar // Proceedings of the 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA); — Penang: IEEE, 2018. — p. 135-139.

7. Miller G. Riemann’s Hypothesis and Tests for Primality. / G. Miller // Journal of Computer and System Sciences. — 1976. — 13. — p. 300-317.

8. Rabin M.O. Probabilistic Algorithm for Testing Primality. / M.O. Rabin // Journal of Number Theory. — 1980. — 12. — p. 128-138.

9. Nari K. Strong Pseudo Primes to Base 2 [Electronic source] / K. Nari, E. Ozdemir, N.A. Ozkirisci // arXiv. — 2019. — URL: https://arxiv.org/abs/1905.06447. (accessed: 25.03.23)

10. Sorenson P.J. Two Algorithms to Find Primes in Patterns [Electronic source] / P.J. Sorenson, J. Webster // arXiv. — 2019. — URL: https://arxiv.org/abs/1807.08777. (accessed: 25.03.23)

11. Ribenboim P. The New Book of Prime Number Records / P. Ribenboim — New York: Springer, 1995. — 541 p.

12. Ishmukhametov S.T. On the Number of Witnesses in the Miller–Rabin Primality Test. / S.T. Ishmukhametov, B.G. Mubarakov // Symmetry. — 2020. — 12.

13. Mubarakov B.G. Effektivnaya otsenka testa prostoti Millera-Rabina naturalnikh chisel [Efficient Evaluation of the Miller-Rabin Primality Test of Natural Numbers] / B.G. Mubarakov // Materials of the XIX All-Russian Youth Scientific School-Conference; — Issue 59. — Kazan, 2020. — p. 106-109. [in Russian]

14. Mubarakov B.G. On the Number of Primality Witnesses of Composite Integers. / B.G. Mubarakov // Russian Mathematics. — 2021. — 65. — p. 73-77.

15. Zhumaniezov A.R. The Problem of Error Frequency Distribution in the Miller-Rabin Test For Tripleprime Numbers. / A.R. Zhumaniezov // Proceedings of the 10th International Conference on Foundations of Computer Science & Technology; — Issue 12. — Zurich: AIRCC Publishing Corporation, 2022.

16. Zhumaniezov A.R. Estimating the Distribution of Witnesses of the Primality of the Miller-Rabin Test. / A.R. Zhumaniezov // International Journal on Computational Science & Applications. — 2022. — 12.

17. Bleichenbacher D. Efficiency and Security of Cryptosystems Based on Number Theory dis....of PhD in Engineering: - : defense of the thesis 1996-01-01 : approved 1996-01-01 / D. Bleichenbacher — Zurich: 1996. — 98 p.

18. Zhumaniyozov A.R. Ocenka veroyatnosti oshibki testa Millera-Rabina na poluprosty'x chislax [Estimating the Probability of an Error in the Miller-Rabin Test on Semiprime Numbers]. / A.R. Zhumaniyozov // Proceedings of the International Youth Scientific Forum "LOMONOSOV-2022"; — MOSCOW: MAKS PRESS, 2022. [in Russian]

19. Zhang Z. Two Kinds of Strong Pseudoprimes up to 10^36. / Z. Zhang // Mathematics of Computation. — 2007. — 76. — p. 2095-2107.

20. Sorenson J. Strong Pseudoprimes to Twelve Prime Bases. / J. Sorenson, J. Webster // Mathematics of Computation. — 2015. — 86. — p. 985-1003.