On Miller-Rabin Primality Test and Its Worst Witnesses

Jeremy Syaloom Okey Nathanael Simbolon - 13520042[1]
Informatics Study Program
School of Electrical Engineering and Informatics
Bandung Institute of Technology, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13520042@std.stei.itb.ac.id

***Abstract*—The Miller-Rabin primality test is one of the most well-known methods of primality testing used in modern day cryptography. Being wildly used due to its speed and simplicity, extensive effort has been made to optimize it, with many focused on finding its best primality witnesses. Despite such coverage, not much research wildly available have focused on its worst witnesses. In this paper, I use an algorithmic approach to present some of the test's worst witnesses, hoping to encourage more discussions in this topic.**

***Keywords*—Miller-Rabin, number theory, primality test, witness**

I. Introduction

For millennia, human's advancement in science and technology cannot be separated from our improved understanding of pure and applied mathematics. Specifically speaking, the field of number theory has seen tremendous development over the course of our history. From the use of sexagesimal number system by the Babylonians in the year 3000 BC [1] to the proof of Fermat's Last Theorem in 1995 [2], our curiosity and hunger of knowledge gives birth to the mathematical tools we depend on in our modern society. One of such is the field of arithmetic and prime numbers as its foundation.

Prime numbers hold an essential role in today's world. In biology, past publication has shown naturally existing link between prime numbers and the life cycle of insects that are part of the genus *Magicicada* as an evolutionary advantage [3]. Prime numbers also serve as the basis of modern error detection, hash table creation, and cryptography. Due to these, the search and verification of prime numbers, both new and existing ones, are very important to number theory and cryptography especially.

There are several methods that can be used to verify the primality of a number. One of those is the Miller-Rabin primality test, first discovered by Gary L. Miller in 1976 [4] and modified by Michael O. Rabin in 1980 [5] to become the probabilistic version widely used today [6]. Due to its popularity, researchers have put enormous efforts to optimize this test. Past publications have presented optimal choices of witnesses to increase the efficiency of this test method [7] – [9]. However, there is little to no publication widely available and freely accessible that is focused on discussing the test's worst witnesses, i.e., those who act as a strong liar to the strong pseudoprime it is testing.

In this paper, I shall present an algorithmic way to showcase the test's worst witnesses. Hopefully, this paper will induce more discussion and further research in this relatively unexplored topic.

II. Theoretical Foundation

A. Fundamentals of Number Theory

Number theory is the branch of pure mathematics that is devoted to the study of integers, primarily positive integers (also known as natural numbers), and the relationship between its member [10]. The history of number theory dates back to the year 3000 BC in the form of a number system that influenced how we interpret time to this day [1]. In modern day, discoveries and further understanding of this field lead to its broad application in geometry, statistics, biology, physics, computer science, and cryptography, to name a few. In the following sections, several key ideas in number theory will be presented.

Divisibility. — Let a and b be integers, $b \neq 0$. We say that a *divides* b (or b is *divisible* by a) if and only if there exists an integer d such that

$$a = bd. \tag{1}$$

In mathematical notation, we can express the above statement into the following form.

$$a \mid b. \tag{2}$$

Let c be an integer *not divisible* by a. We can denote that statement into the following notation.

$$a \nmid c. \tag{3}$$

The following facts are true for $a, b, c \in \mathbb{Z}$ [11].

1) $a \mid a, 1 \mid a,$ and $a \mid 0$;
2) $a \mid 1$ if and only if $a = \pm 1$;
3) $0 \mid a$ if and only if $a = 0$;
4) $a \mid b$ and $b \mid a$ if and only if $a = \pm b$;
5) $a \mid b$ if and only if $-a \mid b$ if and only if $a \mid -b$;
6) $a \mid b$ and $a \mid c$ implies $a \mid (b + c)$;
7) $a \mid b$ and $b \mid c$ implies $a \mid c$.

Remainders. — We shall expand our previous definition in the realm of divisibility. Let a and b be integers, $b \neq 0$. Then there exist integers q and r such that

$$a = qb + r, \qquad 0 \leq r < |b|. \tag{4}$$

The number q and r is respectively called the *quotient* and the *remainder* of a when divided by b [12]. The remainder r is uniquely determined by the division of a and b. In addition, the remainder r is equal to zero if and only if a is divisible by b.

Greatest common divisor. — Let a and b be integers, not both zero. Then there exists an integer d that satisfy

$$d \mid a, \qquad d \mid b.$$

Such d is called the *common divisor* of a and b. Moreover, we call such d the *greatest common divisor* of a and b if d is nonnegative and all other common divisors of a and b divide d [11]. The greatest common divisor of a and b can be written into the following notation.

$$\gcd(a, b)\,. \tag{5}$$

Finding the gcd of two integers can be done efficiently using the Euclidean algorithm, first described by the ancient Greek mathematician Euclid in his treatise, *The Elements*, back in 300 BC [13]. For two integers a and b, the algorithm can be illustrated as follows.

1) Picking a as the larger of the two integers, write a in the form described in (4).
2) Choose b as the new a and r as the new b, then repeat the first step.
3) Proceed until r equals to zero, then r from the previous recursion is our answer.

The algorithm can be written in Python as the following recursive code.

```
def gcd(a, b):
    if b == 0:
        return a
    else
        return gcd(b, a % b)
```

Bézout's identity. — Let a, b, and r be integers and let d be the gcd of a and b. Then, there exist integers s and t such that the following is satisfied.

$$as + bt = r, \qquad d \mid r. \tag{6}$$

The statement above is called *Bézout's identity*, named after the French mathematician, Étienne Bézout, who proved the polynomial version of this identity in 1779 [14]. In the statement, s and t are called the *Bézout's coefficient* of pairs (a, b), which is not unique for that pair. As an example, both the following are true.

$$12 = 24(-1) + 36(1)$$
$$12 = 24(-4) + 36(3)$$

B. Modular Arithmetic

Modular arithmetic is the branch of arithmetic that concerns the relationship of divisibility between two integers [14]. Modular arithmetic was developed by the German mathematician Karl Friedrich Gauss and first presented in his book *Disquisitiones Arithmeticae* in 1801. Other than number theory itself, applications of modular arithmetic has been shown in the field of computer science, chemistry, and music. In the following sections, several key ideas in modular arithmetic will be presented.

Congruences. — Let m be a positive integer. If a and b are integers, we say that a is *congruent* to b *modulo* m if

$$m \mid (a - b). \tag{7}$$

The integer m is called the *modulus* of the congruence. The above statement can be rewritten in the form of (4) as follows.

$$a = km + b. \tag{8}$$

In mathematical notation, we can express the above statement into the following form.

$$a \equiv b \,(mod\ m). \tag{9}$$

Let c be an integer such that $m \nmid (a - c)$. We can state that a is *incongruent* to c *modulo* m and denote that statement into the following notation.

$$a \not\equiv c \,(mod\ m). \tag{10}$$

The following properties are true for $a, b, c, d, k \in \mathbb{Z}, m \in \mathbb{Z}^+$ [14].

1) $a \equiv a \,(mod\ m)$;
2) $a \equiv b \,(mod\ m)$ implies $b \equiv a \,(mod\ m)$;
3) $a \equiv b \,(mod\ m)$ and $b \equiv c \,(mod\ m)$ implies $a \equiv c \,(mod\ m)$;
4) $a \equiv b \,(mod\ m)$ implies $a + c \equiv b + c \,(mod\ m)$;
5) $a \equiv b \,(mod\ m)$ implies $a - c \equiv b - c \,(mod\ m)$;
6) $a \equiv b \,(mod\ m)$ implies $ac \equiv bc \,(mod\ m)$;
7) $a \equiv b \,(mod\ m)$ implies $a^k \equiv b^k \,(mod\ m)$;
8) $a \equiv b \,(mod\ m)$ and $c \equiv d \,(mod\ m)$ implies $a + c \equiv b + d \,(mod\ m)$;
9) $a \equiv b \,(mod\ m)$ and $c \equiv d \,(mod\ m)$ implies $a - c \equiv b - d \,(mod\ m)$;
10) $a \equiv b \,(mod\ m)$ and $c \equiv d \,(mod\ m)$ implies $ac \equiv bd \,(mod\ m)$;

Linear congruences. — Let m be a positive integer. If a and b are integers, a congruence in the form

$$ax \equiv b \,(mod\ m) \tag{11}$$

with x being an unknown integer is called a *linear congruence in one variable* [14]. Let $d = \gcd(a, m)$. If $d \nmid b$, then (11) has no solutions. Conversely, if $d \mid b$, then (11) has exactly d incongruent solutions, i.e., having a solution set of d integers which are incongruent with each other in modulo m. The

solution x is given by

$$x = \frac{km + b}{a}, \qquad k \in \mathbb{Z}. \tag{12}$$

Note that if $\gcd(a, m) = 1$, (11) has a unique solution.

Modular inverses. — We shall now discuss the linear congruence in the special form

$$ax \equiv 1 \ (mod\ m), \tag{13}$$

where a is an integer, m is a positive integer, and x being an unknown integer. As previously stated, there is a unique solution to (13) if and only if $\gcd(a, m) = 1$. Such x is then called the *inverse of* a modulo m. When we obtain the value of x, we can use it to solve any linear congruence in the form of (11). This can be done by letting $\bar{a}$ be the inverse of a modulo m. Then, we can multiply both sides by $\bar{a}$ to get

$$x \equiv \bar{a}b \ (mod\ m). \tag{14}$$

C. Prime Numbers

We have previously stated that for an integer a, it is true that $1 \mid a$ and $a \mid a$. We can then conclude that every integer larger that 1 has at least two positive divisors, 1 and the number itself. We shall then shift our attention to focus on positive numbers who only have exactly two divisors. Such numbers are called *prime numbers*. We shall call other integers larger than 1 that is not prime as *composite numbers*. The first well documented study of prime numbers was done by Euclid in 300 BC, as recorded in his treatise, *The Elements* [14]. Along with his study, Euclid also proved that there exists an infinite number of prime numbers. As of today, prime numbers hold an important role in algebra, geometry, biology, computer science, and cryptography. In the following sections, several key ideas involving prime numbers will be presented.

The Fundamental Theorem of Arithmetic. — The fundamental theorem of arithmetic highlights the importance of prime numbers being the building block of other positive integers. The modern statement of the theorem can be stated as follows [14].

> *Every positive integer greater than 1 can be uniquely written as a product of prime numbers, with the prime factors in the product written in nondecreasing order.*

The prime power factorization of a positive integer a encodes essential information about the number. With the given factorization, one can deduce whether a prime number p divides a simply by checking the appearance of p in the prime factorization of a.

Prime relativity. — Let a and b be positive integers. we say a and b are *relatively prime* or *coprime* if and only if the only positive divisor for both integers is 1. Equivalently,

$$\gcd(a, b) = 1. \tag{15}$$

It follows that there exist integers s and t such that the Bézout's identity is fulfilled for integers a and b. Expressing the statement in the form presented in (6),

$$as + bt = 1. \tag{16}$$

It also follows that in (13), x has a unique solution if a and m are relatively prime.

D. Fermat's Little Theorem

Fermat's Little Theorem, sometimes also being referred as Fermat's Theorem, is one the most fundamental discovery in the field of number theory. The theorem is first stated by Pierre de Fermat, a French mathematician, to a fellow French mathematician and friend, Bernard Frénicle de Bessy, in 1640 [15]. At the time, Fermat did not provide a proof to the theorem. The first published proof of this theorem is provided by Euler in 1736. The theorem plays an important role in deciding whether a given number is prime or not, particularly those that are large. Fermat's Little Theorem is widely used in cryptography, especially public-key cryptography like RSA.

The theorem shall be stated as follows. "*Let p be a prime and a be an integer not divisible by p. Then $a^{p-1} - 1$ is divisible by p.*" The theorem can also be written as

$$a^{p-1} \equiv 1 \ (mod\ p). \tag{17}$$

A direct consequence of the theorem is the following. "*Let p be a prime and a be an integer. Then $a^p - a$ is divisible by p.*" The previous can also be written as the following.

$$a^p \equiv a \ (mod\ p). \tag{18}$$

It should be noted that the converse of the theorem is not guaranteed to be true. If a and p are relatively prime and satisfy (17), p need not to be prime. If it is not, then p shall be called a *pseudoprime* to base a.

E. Miller-Rabin Primality Test

The Miller-Rabin primality test (MRPT) is a probabilistic primality test in the form of an algorithm used to determine whether a given number is likely prime or not. It is first presented as a deterministic primality test by Gary L. Miller in 1976 [4]. However, the version Miller proposed is dependent on the Extended Riemann Hypothesis, a classic mathematical conjecture yet to be proven. In 1980, Michael O. Rabin proposed an improvement of Miller's work which removed the previously needed dependency in the form of a probabilistic primality test [5]. The MRPT is extensively used in the field of cryptography, especially in RSA cryptography to generate large prime numbers [16], [17]. In the following section, we shall present the probabilistic version of the test.

Probabilistic MRPT. — The probabilistic test proposed by Miller and Rabin can be described as follows.

1) Let n be an odd positive integer greater than 2.
2) Write n in the form of $(2^m \times d) + 1$ where d is a positive odd integer.

3) Pick a random integer a between 2 and $n-1$.
4) Check whether $a^d \equiv \pm 1 \ (mod\ n)$.
5) If step 4 is not satisfied, we can conclude that n is composite. Otherwise, n is likely to be prime and step 3 shall be repeated with a different integer a.
6) The process may be repeated until all possible a has been tested or some number of a have been picked such that we can conclude n is prime within a targeted error bound.

The MRPT will not report any false negatives, i.e., if n is a prime, n will pass the test. However, the MRPT will sometimes report false positives, i.e., n may pass the test if n is a strong pseudoprime to base a. The error bound of the MRPT is at most $\left(\frac{1}{4}\right)^k$, where k is the number of iterations being performed.

III. Methodology

We shall determine the worst witnesses of the MPRT by figuring out which integer(s) produce the greatest number of false positives compared to other witnesses. To accomplish that, we shall pick each possible odd composite number $n \in [2, k], k \in \mathbb{Z}^+$ and test its primality using every possible witness $a \in (1, n)$. Because every n is a composite number, a should not return a positive result. If a composite number a claims that n is a prime, we can conclude that it is a false positive. We shall then tally the number of false positives for each possible a.

I have written a short Python code that can be used to achieve our goal. The full source code and a copy of this paper can be found in https://github.com/tastytypist/miller-rabin. Below is a snippet of the source code used for this paper.

```
witness_fail_count = {}
d_values = {}

def witness_check(a, n):
    if n in d_values:
        d = d_values[n]
    else:
        d = n - 1
        while d % 2 == 0:
            d //= 2
        d_values[n] = d

    if (a ** d) % n == 1 or (a ** d) % n == n - 1:
        if a in witness_fail_count:
            witness_fail_count[a] += 1
        else:
            witness_fail_count[a] = 1
```

In the code snippet above, I utilize hash tables `witness_fail_count` and `d_values` to store the false positive tally and the d value previously calculated to increase the efficiency of the code. The function `witness_check` shall accept integers a and n as its parameters. The value of a can be generated using a for-loop that iterates between 2 and n inclusive, and the value of n can be fetched from a tuple of composite numbers available in the full source code. We shall then update the tally hash table if the value pair a and n satisfy step 4 of the probabilistic MRPT, indicating false positive.

IV. Results

Here are the worst witnesses of MRPT for $k = 100, 1000,$ and 10000 respectively.

Table I. The five worst witnesses for $k = 100$

Witness	False positives
38	2
62	2
74	2
18	1
19	1

Table II. The ten worst witnesses for $k = 1000$

Witness	False positives
64	7
230	7
256	7
30	6
74	6
16	6
81	6
149	6
374	6
373	6

Table III. The twenty worst witnesses for $k = 10000$

Witness	False positives
256	53
16	36
4096	34
1296	29
64	28
529	23
729	23
1024	23
625	21
900	21
81	20
373	20
1451	20
75	19
300	19
1156	19
100	18
223	18
484	18
676	18

The full breakdown of the result is also available at the GitHub repository of this paper, accessible via the following link: https://github.com/tastytypist/miller-rabin.

V. Discussion

As we can see in Table I, Table II, and Table III, for $k = 100$, the five worst witnesses a are 38, 62, 74, 18, and 19. For $k = 1000$, the ten worst witnesses a are 64, 230, 256, 30, 74, 16, 81, 149, 374, and 373. For $k = 10000$, the twenty worst witnesses a are 256, 16, 4096, 1296, 64, 529, 729, 1024, 625, 900, 81, 373, 1451, 75, 300, 1156, 254, 100, 223, and 484.

Looking at the results, we may notice some patterns that arises when looking at each witness. Some witnesses that have a tendency of returning false positives can be expressed in the form of even powers of prime as follows.

$$64 = 2^6$$
$$256 = 2^8$$
$$16 = 2^4$$
$$81 = 3^4$$
$$4096 = 2^{12}$$
$$529 = 23^2$$
$$729 = 3^6$$
$$1024 = 2^{10}$$
$$625 = 5^4$$

Looking further, some "trigger-happy" witnesses can be expressed as the multiplication of two even powers of prime as follows.

$$1296 = 2^4 \times 3^4$$
$$1156 = 2^2 \times 17^2$$
$$100 = 2^2 \times 5^2$$
$$484 = 2^2 \times 11^2$$
$$676 = 2^2 \times 13^2$$
$$900 = 2^2 \times 3^2 \times 5^2$$

Unfortunately, I am not knowledgeable enough in this topic to make an educated reasoning and conclusion as of why such interesting patterns emerge. Furthermore, stating a wild guess in this paper simply seems reckless and unwise. I believe it is in my best interest to refrain from doing just that.

VI. Conclusion

The lack of existing publication discussing the worst witnesses of the Miller-Rabin primality test motivates me to write this paper to do exactly just that. In this paper, I have presented the worst witnesses of the Miller-Rabin primality test. I have also argued that there might exist a pattern for the witnesses who raise more false positives compared to other witnesses. Unfortunately, I am not well versed enough in this topic to give any good explanation as of why that is the case. Hopefully, this paper may spark more discussion regarding this topic and potentially help the discovery of future knowledge in the realm of number theory.

As a suggestion for future researchers who are interested in this topic regarding worst witnesses, it may be advisable to present the result in the form of accuracy percentage, noting that smaller values of a is being tested more often compared to a significantly larger a. Although this might not change the conclusion significantly at first glance, such change in method may result in a better understanding of the data.

VII. Acknowledgment

This paper would not have been possible without the opportunities given by my lecturer, Dr. Ir. Rinaldi Munir, M.T. His enthusiasm, curiosity, and ability to bring out the best of his student motivated me to explore and push my limits in my study. I would like to also thank Mr. Matt Parker as the sole inspiration for picking this topic.

References

[1] Øystein Ore, *Number Theory and Its History*. New York: Dover Publucations, 1988.

[2] A. Wiles, "Modular Elliptic Curves and Fermat's Last Theorem," *The Annals of Mathematics*, vol. 141, no. 3, p. 443, May 1995, doi: 10.2307/2118559.

[3] P. R. A. Campos, V. M. de Oliveira, R. Giro, and D. S. Galvão, "Emergence of Prime Numbers as the Result of Evolutionary Strategy," *Physical Review Letters*, vol. 93, no. 9, Aug. 2004, doi: 10.1103/physrevlett.93.098107.

[4] G. L. Miller, "Riemann's Hypothesis and tests for primality," in *Proceedings of seventh annual ACM symposium on Theory of computing - STOC '75*, 1975, pp. 234–239, doi: 10.1145/800116.803773.

[5] M. O. Rabin, "Probabilistic algorithm for testing primality," *Journal of Number Theory*, vol. 12, no. 1, pp. 128–138, Feb. 1980, doi: 10.1016/0022-314x(80)90084-0.

[6] S. Ishmukhametov and B. Mubarakov, "On Practical Aspects of the Miller-Rabin Primality Test," *Lobachevskii Journal of Mathematics*, vol. 34, no. 4, pp. 304–312, Oct. 2013, doi: 10.1134/s1995080213040100.

[7] G. Jaeschke, "On Strong Pseudoprimes to Several Bases," *Mathematics of Computation*, vol. 61, no. 204, pp. 915–926, 1993, doi: 10.2307/2153262.

[8] Y. Jiang and Y. Deng, "Strong Pseudoprimes to the First Eight Prime Bases," *Mathematics of Computation*, vol. 83, no. 290, pp. 2915–2924, May 2014, doi: 10.1090/s0025-5718-2014-02830-5.

[9] J. Sorenson and J. Webster, "Strong Pseudoprimes to Twelve Prime Bases," *Mathematics of Computation*, vol. 86, no. 304, pp. 985–1003, Jun. 2016, doi: 10.1090/mcom/3134.

[10] J. H. Silverman, *A Friendly Introduction to Number Theory*. Boston: Pearson, 2013.

[11] V. Shoup, *A Computational Introduction to Number Theory and Algebra*. Cambridge: Cambridge University Press, 2009.

[12] L. K. Hua, *Introduction to Number Theory*. Berlin: Springer Berlin Heidelberg, 1982.

[13] K. H. Rosen, *Discrete Mathematics and Its Applications*. New York: Mcgraw-Hill, 2019.

[14] K. H. Rosen, *Elementary Number Theory*. Boston: Pearson Education, 2011.

[15] D. M. Burton, *The History of Mathematics: An Introduction*. New York: McGraw Hill, 2011.

[16] D. Apdilah, N. Khairina, and M. K. Harahap, "Generating Mersenne Prime Number Using Rabin Miller Primality Probability Test to Get Big Prime Number in RSA Cryptography," *International Journal Of Information System & Technology*, vol. 1, no. 1, pp. 1–7, Nov. 2017, doi: 10.30645/ijistech.v1i1.1.

[17] G. Dordevic and M. Markovic, "On Optimization of Miller-Rabin Primality Test on TI TMS320C54x Signal Processors," *2007 14th International Workshop on Systems, Signals and Image Processing and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*, 2007, pp. 229-232, doi: 10.1109/IWSSIP.2007.4381195.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bogor, 13 Desember 2021

Jeremy S.O.N. Simbolon
13520042