Akdeniz University Journal of Science and Engineering
Akd.Uni. J. Sci.Eng.

2024 - 01, pp. 08-18

PROBABILISTIC PRIMALITY TESTS AND RSA ALGORITHM

Fatma ÇETİN[1]* **Ahmet SINAK[2]**

[1] Institute of Science, Necmettin Erbakan University, 42090 Konya, Türkiye
[2]Department of Management Information Systems, Akdeniz University, 07600 Antalya, Türkiye

ABSTRACT

The security of the RSA algorithm is based on the difficulty of the integer factorisation problem. Two large prime numbers are needed to construct an RSA algorithm for each user. This leads to the issue of generating large prime numbers in cryptography. In the literature, there are two main primality test methods: probabilistic and deterministic primality tests. This paper reviews the main probabilistic primality tests such as the Fermat, Slovay-Strassen and Miller-Rabin test algorithms. Then we evaluate and compare their performance based on their execution times for different sizes of inputs. We present performance analyses based on their execution times. We also review the RSA encryption algorithm that uses two sufficiently large prime numbers.

Keywords: *Cryptology, Prime numbers, Probabilistic Primality tests, RSA algorithm*

1. INTRODUCTION

The Integer Factorization Problem (IFP) is assumed as a difficult problem in mathematics for sufficiently large prime numbers. The security of the RSA algorithm is based on the difficulty of the IFP for the product of two large prime numbers. Thus, to ensure the security of the RSA algorithm, sufficiently large prime numbers must be generated. This is a challenging problem in cryptography (indeed, in number theory). In the literature, the current deterministic primality tests are not efficient for large numbers. In this context, the probabilistic primality tests are used to generate large prime numbers for the RSA algorithm.

Prime numbers were first studied in detail by the mathematicians of the Pythagorean school in ancient Greece between 500 - 300 BC. In 200 BC, Eratosthenes developed a method for finding prime numbers and named this method the "Sieve of Eratosthenes." It is a well-known fact that every natural number can be expressed as a product of the powers of prime numbers. Moreover, the number of prime numbers is infinite. In the literature, numerous scientists have studied the characterization of prime numbers and discovered significant results on prime numbers. However, any efficient deterministic primality test algorithm hasn't yet been proposed in the literature to test sufficiently large numbers.

Public key cryptosystems based on prime numbers are frequently used for encryption and signature processes in real life. Sufficiently large prime numbers are required to ensure the security of certain public key cryptosystems. Thus, prime numbers are always needed in cryptography. The mystery of prime numbers, which is still not fully understood, increases interest in mathematics and computer science. Primality tests are among the first studies conducted on prime numbers.

In the main paper [6], the author aims to introduce quite modern cryptography and applications. An algorithmic approach has been emphasized with a focus on efficiency estimates. In the paper [13], a deterministic testing method has been developed to check whether an odd number is prime. In the paper

1)
*Corresponding Author: ffatmactn97@gmail.com
Recieving Date:16.12.2024 Publishing Date: 30.12.2024

[7], a new method for finding prime numbers has been provided and a perfect secure prime number sequence has been defined. In the paper [12], it has been observed that if Michael O. Rabin's primality test failed with a 25% probability on every composite number, factoring would be easy. The reliability of Rabin's test when used to generate a random integer that is probably prime, rather than testing a specific number for primality, is also among the topics of research.

The paper is organised as follows. In Section 2, the probabilistic primality tests, including the Fermat, Solovay-Strassen and Miller-Rabin tests are discussed. These tests allow us to determine whether an odd number is composite or prime with high probability. In Section 3, we address the security of the RSA algorithm based on two large prime numbers. In Section 4, the performance analyses of the probabilistic primality tests are provided in terms of their running time.

2. PROBABILISTIC PRIMALITY TESTS

In this section, we discuss the probabilistic primality tests such as the Fermat, Slovay-Strassen and Miller-Rabin tests.

Prime numbers and their properties were first studied in detail by mathematicians of the Pythagorean school in ancient Greece between $\mathbf{500-300}$ BCE. In $\mathbf{200}$ BCE, Eratosthenes developed the sieve method to find prime numbers, named the "Sieve of Eratosthenes." The Sieve of Eratosthenes is a method used to find prime numbers up to a certain integer. However, this method is not practical to test very large numbers, which is why it is not used in cryptography. Therefore, in cryptography, probabilistic primality tests are used to test sufficiently large prime numbers.

Probabilistic Primality Tests are used to test whether an odd number is composite or prime with high probability. The most commonly used probabilistic primality tests are the Fermat, Solovay-Strassen, and Miller-Rabin tests.

We first provide some probabilistic definitions before introducing the probabilistic primality tests. The probabilistic primality test is based on the concept of a witness and a liar.

Definition 1. [13] For a number n, if there is a number a between 1 and $n-1$ such that a confirms that n is a composite number according to the test, then a is called *witness* for the composite number n. On the other hand, there may exist a number a that says a composite number n may be prime. Such a number a is called a *liar* for a composite number n.

Note that when the liar a is used in the test, the test will incorrectly declare a composite number n to be prime. To avoid such errors, repeating the test t times (for a sufficiently large value t) will further reduce the probability of error.

2.1. Fermat's Probabilistic Primality Test

The Fermat Probabilistic Primality Test is the first test that forms the basis of probabilistic primality tests. It is based on Fermat's little theorem, which was proposed by Fermat in 1640. Fermat's little theorem can be stated as follows.

Theorem 1. [6] (Fermat's little theorem) If p is an odd prime number and if a is any integer which is not a multiple of p, then we have the congruence

$$a^{p-1} \equiv 1 \bmod p \tag{1}$$

Usually, we assume that $1 \leq a \leq p-1$. For $a = 1$ and $a = p-1$, it is trivial that $a^{p-1} \equiv 1 \bmod p$. Thus, in the Fermat test, we assume that $2 \leq a \leq p-2$.

The equivalent statement of Theorem 1: if $a^{p-1} \not\equiv 1 \bmod p$ for at least one base a with $2 \le a \le p-2$, then p is not a prime (namely, a composite number). On the other hand, if $a^{p-1} \equiv 1 \bmod p$ for some base number a with $2 \le a \le p-2$, then p may still be a prime or composite number. In this case, we cannot say that p is an odd prime number, but we call p as a pseudoprime number with a base a.
The Fermat probabilistic primality test is based on Fermat's little theorem. For simplicity, we refer to the Fermat test. According to the above observation, we below define the Fermat test.

Fermat Test: Let $n \ge 3$ be an odd integer, pick randomly some number a with $2 \le a \le n-2$.

If the congruence $a^{n-1} \not\equiv 1 \bmod n$, then return "$n$ is composite," else return "n is pseudoprime base a".

In the Fermat test, the congruence in (1) is checked for t different values of base a with $2 \le a \le n-2$ to determine whether the number n is a composite or pseudoprime number with a certain error rate. The algorithm of the Fermat test is given below for an odd number n.

Algorithm 1. Fermat's Test Algorithm
Input: n and $t \in \mathbb{Z}^+$
Output: n is a composite or a pseudoprime with the error rate $E_n\ (t)$.
1: **For** pick randomly an integer a with $2 \le a \le n-2$
2: $d \leftarrow \gcd(a, n)$
3: **if** $d > 1$ **return** "composite"
4: **else** $b \leftarrow a^{n-1} \bmod n$
5: **end if**
6: **if** $b \neq 1$ **return** "composite"
7: **end if**
8: **end for**
9: **return** n is a pseudoprime with the error rate $E_n\ (t)$

Example 1. We verify whether 571 is composite or pseudoprime by the Fermat test.
Input: $n = 571$ and $t = 3$ iterations.
1: **For** pick randomly an integer a with $2 \le a \le 569$
2: For $a = 2$, $a^{n-1} = 2^{570} \equiv 1 \bmod 571$
3: For $a = 42$, $a^{n-1} = 42^{570} \equiv 1 \bmod 571$
4: For $a = 123$, $a^{n-1} = 123^{570} \equiv 1 \bmod 571$
Output: 571 is a pseudoprime number.

Definition 2. Let n be an odd composite integer and a be an integer with $1 \le a \le n-1$.
- An integer a with $2 \le a \le n-2$ is called a *Fermat witness* if $a^{n-1} \not\equiv 1 \bmod n$. In other words, an integer a approves that n is composite.
- An integer a with $1 \le a \le n-1$ is a *Fermat liar* for n if $a^{n-1} \equiv 1 \bmod n$.

Definition 3. (Carmichael Numbers**)** In the Fermat primality test, some composite numbers can give misleading results. These composite numbers pass the Fermat primality test for any base although they are not prime numbers. These numbers are called Carmichael numbers. Initially, in 1910, R. D. Carmichael discovered such numbers.

According to Fermat's little theorem, for n to be a prime number, for every base a, $a^n - a$ must divide a. However, there are composite Carmichael numbers that satisfy this division. Therefore, the Fermat test fails to detect Carmichael numbers.

Example 2. We verify whether 561 is a pseudoprime or composite by the Fermat test.
Input: $n = 561$ with $t = 5$ iterations.

1: For pick randomly an integer a with $2 \leq a \leq 559$
2: For $a = 13$, $a^{n-1} = 13^{560} \equiv 1 \pmod{561}$
3: For $a = 29$, $a^{n-1} = 3^{560} \equiv 1 \pmod{561}$
4: For $a = 52$, $a^{n-1} = 52^{560} \equiv 1 \pmod{561}$
5: For $a = 76$, $a^{n-1} = 76^{560} \equiv 1 \pmod{561}$
6: For $a = 125$, $a^{n-1} = 125^{560} \equiv 1 \pmod{561}$
Output: 561 is a pseudoprime number

However, $561 = 3 \cdot 11 \cdot 17$ is a composite number. The base numbers $a = 13, 29, 52, 76$ and 125 are Fermat liars for the composite number 561. Such numbers are called Carmichael numbers.

2.2. Solovay-Strassen Primality Test

The Solovay-Strassen primality test, developed by Robert Solovay and Volker Strassen, is the first probabilistic primality test used in Public Key Cryptography. This test is based on the Jacobi symbol and Euler's criterion. The Jacobi symbol is a generalization of the Legendre symbol, introduced by Jacobi in 1837.

Jacobi Symbol. [12] Given any positive odd integer $\boldsymbol{n}$ and any integer $\boldsymbol{a}$, the Jacobi symbol $\left(\frac{\boldsymbol{a}}{\boldsymbol{n}}\right)$ is defined as

$$\left(\frac{a}{n}\right) = \begin{cases} 1 \text{ if } a \text{ is a quadratic residue mod } n \\ -1 \text{ if } a \text{ is a quadratic non} - \text{residue mod } n \\ 0 \text{ if } a \text{ divides } n \end{cases}$$

Theorem 2. (Euler's Criterion) If $\boldsymbol{p}$ is an odd prime number and $\boldsymbol{a}$ is a positive integer satisfying $(\boldsymbol{a}, \boldsymbol{p}) = \boldsymbol{1}$, then the following congruence holds:

$$\boldsymbol{a^{(p-1)/2} \equiv \left(\frac{a}{p}\right) \bmod p}$$

Equivalently, if this congruence does not hold, then $\boldsymbol{p}$ is a composite number. On the other hand, if this congruence holds for at least one base $\boldsymbol{a}$, then $\boldsymbol{p}$ is pseudoprime for base $\boldsymbol{a}$.

According to these observations, the Solovay-Strassen primality test is defined as follows.

Solovay-Strassen Test [12]: Let $\boldsymbol{n}$ be an odd number and $\boldsymbol{a}$ be a number with $\boldsymbol{1 \leq a \leq n-1}$. If

$$\boldsymbol{a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \bmod n}$$

then $\boldsymbol{n}$ is called *pseudoprime* with the base $\boldsymbol{a}$. Otherwise, $\boldsymbol{n}$ is a composite number.

Definition 4. Let $\boldsymbol{n}$ be an odd composite number and $\boldsymbol{a}$ is a number in the range $\boldsymbol{1 \leq a \leq n-1}$.

- If $\boldsymbol{a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) \bmod n}$, $\boldsymbol{a}$ is called an *Euler witness* of $\boldsymbol{n}$.
- If $\boldsymbol{a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \bmod n}$ although $\boldsymbol{n}$ is an odd composite number, $\boldsymbol{a}$ is called an *Euler liar* of $\boldsymbol{n}$.

If n is a prime number, the probability that a is a witness at least 50%. This test is repeated $\boldsymbol{t}$ times using $\boldsymbol{t}$ different values of $\boldsymbol{a}$. The probability of a composite number passing the test for $\boldsymbol{t}$ times are at most $\boldsymbol{\frac{1}{2^t}}$. The algorithm for this test is given below.

Algorithm 2: Solovay-Strassen Test Algorithm
Input: An odd positive integer n and $t \in \mathbb{Z}^+$
Output: n is either composite or pseudoprime with the error rate $E_n(t)$.
1: **For** pick randomly an integer a with $1 \leq a \leq n-1$
2: $d \leftarrow \gcd(a, n)$
3: **if** $d > 1$ **return** "composite"
4: **else** $b \leftarrow a^{\frac{n-1}{2}} \bmod n$
5: **end if**
6: **if** $b \neq \pm 1$ **return** "composite"
7: **end if**
8: $J \leftarrow \left(\frac{a}{n}\right)$
9: **if** $b \neq J \bmod n$ **return** "composite"
10: **end if**
11: **end for**
12: **return** n is pseudoprime with the error rate $E_n(t)$

Example 3. We determine if 349 is composite or pseudoprime by the Solovay-Strassen test.
Input: $n = 349,\ t = 3 \in \mathbb{Z}^+$

1: For $a = 2,\ b = -1 \leftarrow 2^{(349-1)/2} \bmod 349$
2: $J = -1 \leftarrow \left(\frac{2}{349}\right)$
1: For $a = 3,\ b = -1 \leftarrow 3^{(349-1)/2} \bmod 349$
2: $J = -1 \leftarrow \left(\frac{3}{349}\right)$
1: For $a = 5,\ b = -1 \leftarrow 5^{(349-1)/2} \bmod 349$
2: $J = -1 \leftarrow \left(\frac{5}{349}\right)$
Output: 349 is a pseudoprime number.

We finally review the Miller-Rabin probabilistic primality test, which is faster and has a lower error rate compared to the Solovay-Strassen test and the others.

2.3. Miller-Rabin Primality Test

One of the most commonly preferred techniques for testing the primality of a given large odd number is the Miller-Rabin (M-R) probabilistic primality test. This test was developed by Michael Rabin based on the idea of Gary Miller and is particularly known for its low error rate.
In the Miller-Rabin probabilistic test, to determine whether a given odd number $\boldsymbol{n}$ is prime, the first step is to find the values of $\boldsymbol{s}$ and $\boldsymbol{r}$ such that $\boldsymbol{n - 1 = 2^s\, r}$.

Theorem 3. Let $\boldsymbol{p}$ be a positive odd integer and $\boldsymbol{a}$ be a number with $\boldsymbol{1 \leq a \leq p - 1}$. Write $\boldsymbol{p - 1 = 2^s r}$, where $\boldsymbol{r}$ is an odd integer and $\boldsymbol{s}$ is an integer. If $\boldsymbol{p}$ is an odd prime number, then the equation

$\boldsymbol{a^r \equiv 1 \bmod p}$ holds or the equation $\boldsymbol{a^{2^j r} \equiv -1 \bmod p}$ holds for any $\boldsymbol{j}$ with $\boldsymbol{0 \leq j \leq s - 1}$.

Equivalently, if the equation $\boldsymbol{a^r \not\equiv 1 \bmod p}$ and the equation $\boldsymbol{a^{2^j r} \not\equiv -1 \bmod p}$ for every $\boldsymbol{j}$ with $\boldsymbol{0 \leq j \leq s - 1}$, then $\boldsymbol{p}$ is a composite number. On the other hand, for an integer $\boldsymbol{a}$ in the range $\boldsymbol{1 \leq a \leq p - 1}$, if the equation $\boldsymbol{a^r \equiv 1 \bmod p}$ holds, or if for $\boldsymbol{0 \leq j \leq s - 1}$, the equation $\boldsymbol{a^{2^j r} = 1 \bmod p}$ holds, then $\boldsymbol{p}$ is considered as a pseudoprime for the base $\boldsymbol{a}$.

In view of the above observations, one can check whether a positive odd integer $\boldsymbol{n}$ is prime. This method is called the Miller-Rabin test.

Miller-Rabin Test [5]: Let $\boldsymbol{n}$ be a positive odd integer and $\boldsymbol{a}$ be a number with $\mathbf{1} \leq \boldsymbol{a} \leq \boldsymbol{n} - \mathbf{1}$. Write $\boldsymbol{n}-\mathbf{1} = \mathbf{2}^{\boldsymbol{s}}\boldsymbol{r}$, where $\boldsymbol{r}$ is an odd integer and $\boldsymbol{s}$ is an integer.

- If the equation $\boldsymbol{a}^{\boldsymbol{r}} \not\equiv \mathbf{1} \bmod \boldsymbol{p}$ and the equation $\boldsymbol{a}^{\mathbf{2}^{\boldsymbol{j}}\boldsymbol{r}} \not\equiv -\mathbf{1} \bmod \boldsymbol{p}$ for every $\boldsymbol{j}$ with $\mathbf{0} \leq \boldsymbol{j} \leq \boldsymbol{s} - \mathbf{1}$, then $\boldsymbol{p}$ is a composite number.
- If $\boldsymbol{a}^{\boldsymbol{r}} \equiv \mathbf{1} \bmod \boldsymbol{n}$ or $\boldsymbol{a}^{\mathbf{2}^{\boldsymbol{j}}\boldsymbol{r}} \equiv -\mathbf{1} \bmod \boldsymbol{n}$ holds for any $\boldsymbol{j}$ in the range $\mathbf{0} \leq \boldsymbol{j} \leq \boldsymbol{s} - \mathbf{1}$, then $\boldsymbol{n}$ is called a pseudoprime for the base $\boldsymbol{a}$.

The algorithm of this test is given below.

Algorithm 3. Miller-Rabin Test Algorithm
Input: Positive odd integer n and $t \in \mathbb{Z}^+$
Output : n is either composite or prime with the error rate $E_n(t)$.
1: Write $n - 1 = 2^s r$ where r is an odd integer
2: **for** pick randomly an integer a with $1 \leq a \leq n - 1$
2: $d \leftarrow \gcd(a, n)$
3: **if** $d > 1$ **return** "composite"
4: **else** $b \leftarrow a^r \bmod n$
5: **end if**
6: **if** $b \neq \pm 1$
7: **for** j from 1 to $s - 1$
8: $c \leftarrow a^{2^j r} \bmod n$
9: **if** $c = 1$ **return** "composite"
10: **end if**
11. **end for**
12 **if** $c \neq -1$ **return** "composite"
13: **end if**
14: **end if**
15.**end for**
16: **return** n is a pseudoprime with the error rate $E_n(t)$

Definition 5. Let $\boldsymbol{n}$ be an odd composite number and $\boldsymbol{a}$ is a number in the range $\mathbf{1} \leq \boldsymbol{a} \leq \boldsymbol{n} - \mathbf{1}$. Write $\boldsymbol{n}-\mathbf{1} = \mathbf{2}^{\boldsymbol{s}}\boldsymbol{r}$, where $\boldsymbol{r}$ is an odd integer and $\boldsymbol{s}$ is an integer.

- If the equation $\boldsymbol{a}^{\boldsymbol{r}} \not\equiv \mathbf{1} \bmod \boldsymbol{p}$ and the equation $\boldsymbol{a}^{\mathbf{2}^{\boldsymbol{j}}\boldsymbol{r}} \not\equiv -\mathbf{1} \bmod \boldsymbol{p}$ for every $\boldsymbol{j}$ with $\mathbf{0} \leq \boldsymbol{j} \leq \boldsymbol{s} - \mathbf{1}$, then then $\boldsymbol{a}$ is called a "strong witness" for $\boldsymbol{n}$.
- If $\boldsymbol{a}^{\boldsymbol{r}} \equiv \mathbf{1} \bmod \boldsymbol{n}$ or $\boldsymbol{a}^{\mathbf{2}^{\boldsymbol{j}}\boldsymbol{r}} \equiv -\mathbf{1} \bmod \boldsymbol{n}$ holds for any $\boldsymbol{j}$ in the range $\mathbf{0} \leq \boldsymbol{j} \leq \boldsymbol{s} - \mathbf{1}$ although $\boldsymbol{n}$ is an odd composite number, $\boldsymbol{a}$ is called a strong *liar* of $\boldsymbol{n}$.

Example 4. We apply the Miller-Rabin test to check whether 91 is prime.
Input: $n = 91$
Write $n - 1 = 90 = 2 \cdot 45$, where $s = 1,\ r = 45$
For $a = 2,\ b = a^r = 2^{45} \equiv 57 \bmod 91$
Since $b \neq \pm 1 \bmod 91$, **return** "composite"
Output: 91 is composite

Thus, $a = 2$ is a strong witness. Moreover, we test it for different base numbers.
Input: $n = 91$ and $t = 3$,

Write $n-1=90=2\cdot 45$, where $s=1,\ r=45$
For $a=9,\quad b=a^r=9^{45}\equiv 1 \bmod 91$
For $a=16,\quad b=a^r=16^{45}\equiv 1 \bmod 91$
For $a=75,\quad b=a^r=75^{45}\equiv 1 \bmod 91$
Output: 91 is a pseudoprime number.

For randomly selected values of a in the range $1\le a\le 90$, the result indicated that 91 is a pseudoprime number. Since 91=7*13 is not a prime number, $a=9, a=16,\ a=75$ are strong liars.

3. RSA ALGORITHM

In this section, we review the RSA algorithm as an application of large prime numbers. Whitfield Diffie and Martin Hellman introduced public-key cryptography in 1976. Then, in 1977, Ronald Rivest, Adi Shamir and Leonard Adleman proposed the RSA cryptosystem, which became the most widely used public-key cryptography scheme [10].

In the paper [11], after defining RSA, they discuss how it can be used in the upcoming era of electronic mail. This system is based on the factorisation problem. The security of RSA relies on the difficulty of factoring a large integer that is the product of two sufficiently large prime numbers. The reliability of the algorithm is directly proportional to the size of the prime numbers used; however, due to the modular exponential nature of encryption and decryption processes, it presents time-related disadvantages. The RSA cryptosystem is the most widely used public-key cryptography scheme. Today, RSA is used in many applications such as SSL, S-HTTP, S-MIME, S/WAN, and STT. It is also used in web security certificates for credit card transactions.
In the paper [9], the measurement of the distance between the selected primes p and q for RSA is defined, and applications are provided. In the book [10] the authors explain the most important techniques of modern cryptography. In the paper [7], the author has used the perfect secure prime number sequence defined in a new method for finding prime numbers in the RSA encryption method.

3.1. The Structure of the RSA Algorithm

There are three main components in the RSA algorithm. The first step is to generate a key pair, consisting of a public key and the corresponding private key.

RSA Key Generation

1. Two distinct large prime numbers $\boldsymbol{p}$ and $\boldsymbol{q}$ are generated.
2. The value of $\boldsymbol{n}=\boldsymbol{p}\cdot\boldsymbol{q}$ is calculated.
3. The value of $\boldsymbol{\varphi(n)}=(\boldsymbol{p}-\mathbf{1})\cdot(\boldsymbol{q}-\mathbf{1})$ is calculated.
4. A random number $\boldsymbol{e}$ is selected from $\mathbf{1}<\boldsymbol{e}<\boldsymbol{\varphi(n)}$ such that $\mathbf{gcd}\left(\boldsymbol{e},\boldsymbol{\varphi(n)}\right)=\mathbf{1}$.
5. The value of $\boldsymbol{d}$ is found such that $\boldsymbol{e}\cdot\boldsymbol{d}\equiv\mathbf{1}\ \mathbf{mod}\ \left(\boldsymbol{\varphi(n)}\right)$.

The pairs $(\boldsymbol{n},\boldsymbol{e})$ are the public parameters, and $(\boldsymbol{p},\boldsymbol{q},\left(\boldsymbol{\varphi(n)}\right),\boldsymbol{d})$ are the private parameters. The RSA modulo parameter $\boldsymbol{n}$ is always public. The parameter $\boldsymbol{e}$ is the encryption key and the parameter $\boldsymbol{d}$ is the decryption key.

Below are the steps that person A would follow for RSA encryption to encrypt a message $\boldsymbol{m}$ and send the encrypted message to person B.

RSA Encryption

- The person A obtains the person B's public key, which is the pair $(\boldsymbol{n}, \boldsymbol{e})$.
- The message $\boldsymbol{m}$ is written in the range $[\mathbf{0}; \boldsymbol{n} - \mathbf{1}]$.
- Then, $\boldsymbol{c} \equiv \boldsymbol{m}^{\boldsymbol{e}} \ \mathbf{mod}\ \boldsymbol{n}$ is calculated.
- Finally, A sends the encrypted message $\boldsymbol{c}$ to person B.

The process that person B will perform to decrypt the encrypted message $\boldsymbol{c}$ from person A is outlined below.

RSA Decryption

The person B, who wants to decrypt the encrypted message $\boldsymbol{c}$ sent by person A, uses their private key d to calculate: $\boldsymbol{m} \equiv \boldsymbol{c}^{\boldsymbol{d}} \ \mathbf{mod}\ \boldsymbol{n}$ and thus obtains the message $\boldsymbol{m}$.

3.2. Security of the RSA Algorithm

The security of the RSA algorithm derives from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large prime numbers. RSA, one of the public-key encryption algorithms, uses two different keys. Plaintext encrypted with the public key can only be decrypted with the private key. The security of the RSA algorithm relies on selecting very large prime numbers. To ensure the system's security, it is crucial to generate values for p and q, and thus n, that are resistant to factorization algorithms. Therefore, the parameters p and q should be selected according to certain criteria. The selected parameters provide a security level that is proportional to the size of the n parameter [12].

4. THE PERFORMANCE ANALYSES OF THE PRIMALITY TESTS

In this section, we discuss the performance analyses of the probabilistic primality test algorithms. We implement the algorithms of the probabilistic primality tests given in **Algorithms 1,2** and **3.**

This section aims to perform and compare the performance analyses of probabilistic primality tests. When analysing the performance of these tests, criteria such as runtime, memory requirements, and the number of operations performed are considered. Among the probabilistic primality tests, the three main tests, namely Fermat, Solovay-Strassen and Miller-Rabin are compared, and it is found that the Miller-Rabin test performs better in terms of error rate and runtime. The reason for this is that the Fermat test is weak in detecting Carmichael numbers. The Solovay-Strassen test takes longer due to the increased runtime caused by Jacobi symbol calculations. Additionally, while the Solovay-Strassen test operates with an error rate of $(\mathbf{1/2})^{\boldsymbol{t}}$, the Miller-Rabin test provides more accurate results with an error rate of $(\mathbf{1/4})^{\boldsymbol{t}}$ (see in [12] for more detail).

Below, we compare the performance of the probabilistic Primality Tests (Fermat, Miller-Rabin, and Solovay-Strassen) in terms of runtime for numbers with digit lengths ranging from 2 to 10.

Fermat Test: The Fermat test runtimes for numbers with digit lengths ranging from 2 to 10 are provided in Table 1. Here, the time taken for the largest-digit number $\boldsymbol{p} = \mathbf{2147483647}$ is 7.18 seconds.

Table 1. Fermat Test runtime

FERMAT TEST		
Number of Digits	Mersenne Number	Runtime (s)
2	31	1,84
4	1023	1,84
6	262143	3,15
8	16777215	6,04
10	2147483647	7,18

Solovay-Strassen Test: The Solovay-Strassen Test runtimes for numbers with digit lengths ranging from 2 to 10 are provided in Table 2. Here, the time is taken for the largest-digit number $p = 2147483647$ is 5.62 seconds.

Table 2. Solovay-Strassen Test runtime

SOLOVAY-STRASSEN TEST		
Number of Digits	Mersenne Number	Runtime (s)
2	31	1,84
4	1023	1,84
6	262143	3,15
8	16777215	4,04
10	2147483647	5,62

Miller-Rabin Test: Miller-Rabin Test runtimes for numbers with digit lengths ranging from 2 to 10 are provided in Table 3. Here, the time is taken for the largest-digit number $p = 2147483647$ is 3.22 seconds.

Table 3. Miller-Rabin Test runtime

MILLER-RABIN TEST		
Number of Digits	Mersenne Number	Runtime (s)
2	31	1,67
4	1023	1,80
6	262143	3,00
8	16777215	3,10
10	2147483647	3,22

When we perform the performance analysis for numbers in the range of 20 to 200 digits using the Miller-Rabin test and the Solovay-Strassen test, the data shows that the Miller-Rabin test reaches the result faster.

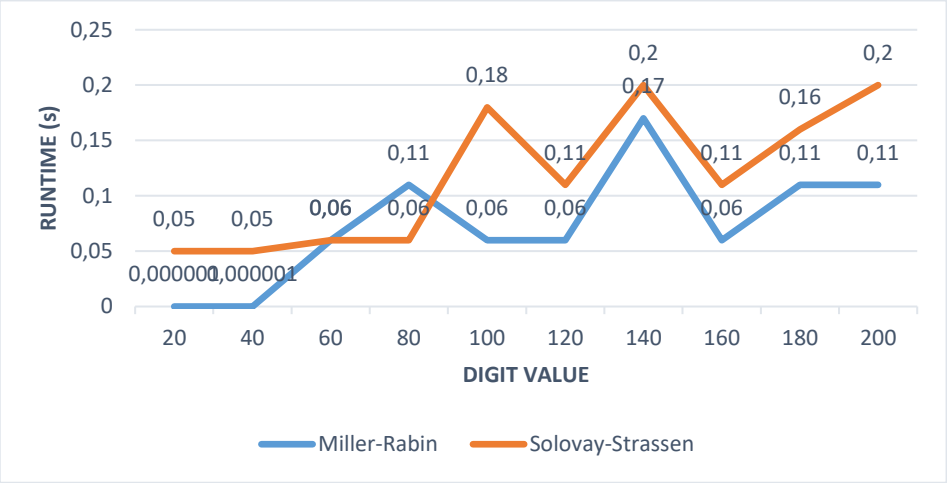

Figure 1. Comparison of the runtime of the Miller-Rabin and Solovay-Strassen tests

5.CONCLUSION

The RSA algorithm is the most popular public-key cryptosystem. This cryptosystem has both encryption and signature algorithms. The security of the RSA cryptosystem is based on the hardness of the integer factorisation problem for two sufficiently large prime numbers. To design the RSA cryptosystem for each person, two sufficiently large prime numbers are needed. Thus, finding sufficiently large prime numbers is a significant problem in the literature. To determine whether large odd numbers are prime, probabilistic primality tests such as Fermat, Solovay-Strassen and Miller-Rabin tests have been examined in this work. Moreover, performance analyses of the Fermat, Solovay-Strassen, and Miller-Rabin tests have been discussed, and their runtimes have been compared. Based on the obtained experimental results, it was concluded that the Miller-Rabin probabilistic primality test is more efficient in terms of speed and performance criteria.

ACKNOWLEDGEMENTS

This work is the output of the Master's thesis in [3] supervised by the second author. We extend our gratitude to Ebru SINAK for her continuous support and contribution to the realization of this work. The first author offers her endless respect and gratitude to her parents, who have always supported her throughout her studies, giving her strength with their presence.

REFERENCES

[1] M. Agrawal, N. Kayal, N. Saxena, *PRIMES is in P*, Annals of Mathematics, **160(2),** 781-793, 2004.

[2] E. Akyıldız, Ç. Çalık, M. Özarar, Z.Y. Tok, O. Yayla, *Security Testing Software for RSA Cryptosystem Parameters*, ISC Turkey 6th International Conference on Information Security and Cryptology, Ankara 2013, p. 124-126, 2013.

[3] F. Çetin, *A Study on Prime Number Test Methods Used in Cryptography*, Master's Thesis, Institute of Science, Necmettin Erbakan University, Under the supervision of Assoc. Prof. Dr. Ahmet Sınak, Konya, 2021.

[4] B.C. Higgins, *The Rabin-Miller Probabilistic Primality Test, Some Results on the Number of Non-Witnesses to Compositeness*, 2000.

[5] N. Koblitz, *A Course in Number Theory and Cryptography*, 2nd Edition, Springer - Verlag, New York, 1994.

[6] N. Koca, *Different Methods and Applications for Prime Number Detection*, Master's Thesis, *Institute of Science, Pamukkale University*, Denizli, 2020.

[7] A. Menezes, P. Van Oorschot, *Handbook of Applied Cryptography*, CRC Press, 1997.

[8] S. Nasibov, *On Cryptographic Systems and Applications*, Master's Thesis, Institute of Science, Ege University, İzmir, 2015.

[9] C. Paar, J. Pelzl, *Understanding Cryptography, A Textbook for Students and Practitioners,* Springer–Verlag, 2009.

[10] R. L. Rivest, A. Shamir, and A. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM, **21(2),** 120–126, 1978.

[11] B. Schneier, *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C* (cloth), John Wiley & Sons Inc, 1996.

[12] A. Segre, *Computer and Network Security*, Data Security, Iowa University, 2000.

[13] T. Yerlikaya, O. Kara, *Prime Number Testing Algorithms Used in Cryptography*, Review Article, Trakya University Journal of Engineering Sciences, **18(1):** 85-94, Edirne, 2017