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

2024, 1(2), 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 factorization problem. Two large prime numbers are required 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 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 to be a difficult problem in mathematics for sufficiently large 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, there are deterministic primality tests such as the AKS primality test, but they are not efficient for large numbers. Thus, the probabilistic primality tests are used to generate large prime numbers for the RSA algorithm and the other public key cryptosystems. Public key cryptosystems based on prime numbers are frequently used for encryption, signature and key-exchange 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.

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." 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. 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 has not yet been proposed in the literature to test sufficiently large numbers. Therefore, in cryptography, probabilistic primality tests are used to test sufficiently large prime numbers.

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

In the literature, several studies are focusing on primality tests for large numbers (for example, [4,6,7,9,10,11,12,13]). The famous textbook [9] presents a comprehensive study of modern cryptography including prime numbers, and their role in cryptography. The other popular textbook [11] discusses different ways of using cryptographic algorithms. Menezes and Oorschot have written the main handbook on modern cryptography and its applications [7]. In this nice book, a new method for finding prime numbers has been provided and a perfect secure prime number sequence has been defined. The thesis [6] investigates different methods for prime number detection. An algorithmic approach has been emphasized focusing on efficiency estimates. In the paper [13], a deterministic testing method has been developed to determine whether an odd number is prime. In the paper [4], the average probability of errors in the Miller-Rabin test is examined, and it is concluded that this probability decreases as the length of the tested numbers increases.

The paper is organised as follows. In Section 2, the probabilistic primality tests such as 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 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 and error rate.

2. PROBABILISTIC PRIMALITY TESTS

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

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

The probabilistic primality test is based on the concept of a witness and a liar. We first provide their definitions.

Definition 1. [13] Let n be a composite number and let a be a number between 1 and $n-1$. If the base a confirms that n is a composite number according to the test, then a is called *witness* for the composite number n. If the base a says that n is probably prime although n is a composite number, then 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 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, we assume that $2 \leq a \leq p-2$ in the test.
The equivalent statement of Theorem 1 is given as follows. If $a^{p-1} \not\equiv 1 \bmod p$ for at least one base a with $2 \leq a \leq p-2$, then p is not a prime (namely, a composite number). Conversely, if $a^{p-1} \equiv 1 \bmod p$ for some base number a with $2 \leq a \leq p-2$, then p may still be a prime or composite number. In this case, we cannot say 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. Because of the above observation, we define the Fermat test as follows.

Fermat Test: Let $n \geq 3$ be an odd integer, pick randomly some number a with $2 \leq a \leq n-2$. 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 $2 \leq a \leq n-2$ to determine whether the number n is a composite or pseudoprime number with a certain error rate $E_n(t)$.

If t different values of base a are randomly selected, there is at most $\frac{1}{2^t}$ probability that the Fermat test will not detect the compositeness of the composite number n. Hence, the probability of a false result in the Fermat test is at most $\frac{1}{2^t}$. This says that the error rate of the Fermat test is defined as $E_n(t) = \frac{1}{2^t}$. For a large enough t, this probability (i.e. error rate) is almost zero.

The algorithm of the Fermat test is given below for an odd number n.

Algorithm 1. Fermat Primality Test Algorithm
Input: An odd integer n and $t \in \mathbb{Z}^+$
Output: n is a composite or a pseudoprime with the error rate $E_n(t)$
1: **For** pick an integer randomly a with $2 \leq a \leq n-2$
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 $E_n(t)$

Below you can find an example of Algorithm 1.

Example 1. We verify whether 571 is composite or pseudoprime by the Fermat test.
Input: $n = 571$ with $t = 3$ iterations.
1: For pick an integer randomly a with $2 \leq a \leq 569$
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 with the error rate $E_n(t) = \frac{1}{2^3}$

Definition 2. Let n be an odd composite number and a be an integer with $1 \leq a \leq n-1$.

- An integer a with $2 \leq a \leq n-2$ is called a *Fermat witness* if $a^{n-1} \not\equiv 1 \bmod n$. An integer a approves that n is composite.

- An integer a with $1 \leq a \leq n-1$ is a *Fermat liar* for n if $a^{n-1} \equiv 1 \bmod n$.

The Fermat primality test gives always misleading results for some composite numbers. These composite numbers are called Carmichael numbers. Initially, in 1910, R. D. Carmichael discovered such numbers.

Definition 3. (Carmichael Numbers**)** If a composite number n passes the Fermat primality test for any base a, then n is called a Carmichael number.

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. More clearly, when we apply the Fermat test to a Carmichael number, the result will always be probably prime. Then the Fermat test will always give a false result for a Carmichael number. The error probability of the Fermat test is virtually 100%.

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 an integer randomly a with $2 \leq a \leq 559$
2: For $a = 13$, $a^{n-1} = 13^{560} \equiv 1 \bmod 561$
3: For $a = 29$, $a^{n-1} = 3^{560} \equiv 1 \bmod 561$
4: For $a = 52$, $a^{n-1} = 52^{560} \equiv 1 \bmod 561$
5: For $a = 76$, $a^{n-1} = 76^{560} \equiv 1 \bmod 561$
6: For $a = 125$, $a^{n-1} = 125^{560} \equiv 1 \bmod 561$
Output: 561 is a pseudoprime number with the error rate $E_n(t) = \frac{1}{2^5}$

Since 561 = 3 · 11 · 17 is a composite number, the bases $a = 13, 29, 52, 76$ and 125 are Fermat liars for the composite number 561. A composite number 561 is a Carmichael number.

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 generalisation 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

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

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. Conversely, if this congruence holds for at least one base $\boldsymbol{a}$, then $\boldsymbol{p}$ is pseudoprime for base $\boldsymbol{a}$.

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

Solovay-Strassen Test: 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.

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 $\frac{1}{2^t}$. The error rate of the Solovay-Strassen test is defined as $\boldsymbol{E_n\,(t) = \frac{1}{2^t}}$. The algorithm of the Solovay-Strassen test is given below.

Algorithm 2: Solovay-Strassen Test Algorithm
Input: An odd integer n and $t \in \mathbb{Z}^+$
Output: n is either composite or pseudoprime with the error rate $E_n(t)$.
1: **For** pick an integer randomly 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 $E_n(t)$

We provide an example of Algorithm 2.

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 $a = 2,\ b = -1 \leftarrow 2^{348/2} \bmod 349$
2: $J = -1 \leftarrow \left(\frac{2}{349}\right)$
3: For $a = 3,\ b = -1 \leftarrow 3^{348/2} \bmod 349$
4: $J = -1 \leftarrow \left(\frac{3}{349}\right)$
5: For $a = 5,\ b = -1 \leftarrow 5^{348/2} \bmod 349$
6: $J = -1 \leftarrow \left(\frac{5}{349}\right)$
Output: 349 is pseudoprime with the error rate $E_n(t) = \frac{1}{2^3}$

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

We finally review the Miller-Rabin probabilistic primality test, which is the fastest 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.

The Miller-Rabin primality test fails with 25% probability on every composite number. This test is repeated t times using t different values of a. The probability of a composite number passing the test for t times is at most $\frac{1}{4^t}$. Then, the error rate of the Miller-Rabin test is defined as $E_n\ (t) = \frac{1}{4^t}$.

In the Miller-Rabin probabilistic test, to determine whether a given odd number $\boldsymbol{n}$ is prime, the first step is to find the number $\boldsymbol{s}$ and odd number $\boldsymbol{r}$ 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 $\boldsymbol{1 \le a \le p-1}$. 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 $\boldsymbol{0 \le j \le s-1}$.

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 $\boldsymbol{0 \le j \le s-1}$, then $\boldsymbol{p}$ is a composite number. Conversely, for an integer $\boldsymbol{a}$ in the range $\boldsymbol{1 \le a \le p-1}$, if the equation $\boldsymbol{a^r \equiv 1 \bmod p}$ holds, or if for $\boldsymbol{0 \le j \le s-1}$, the equation $\boldsymbol{a^{2^j r} = 1 \bmod p}$ holds, then $\boldsymbol{p}$ is considered as a pseudoprime for the base $\boldsymbol{a}$.

Given the above observations, we define the Miller-Rabin primality test based on Theorem 3. One can check whether a positive odd integer $\boldsymbol{n}$ is prime as follows.

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

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

The algorithm of the Miller-Rabin test is given below.

Algorithm 3. Miller-Rabin Test Algorithm
Input: An 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 an integer randomly a with $1 \le a \le n-1$
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 $E_n(t)$

We present an example of Algorithm 3.

Example 4. We apply the Miller-Rabin test to check whether 91 is prime.
Input: $n = 91$ and $t = 1$
Write $n - 1 = 90 = 2 \cdot 45$, where $s = 1$ and $r = 45$
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

Example 5. Check whether 91 is prime by the Miller-Rabin test
Input: $n = 91$ and $t = 3$
Write $n - 1 = 90 = 2 \cdot 45$, where $s = 1$, $r = 45$
For $a = 9$, $b = a^r = 9^{45} \equiv 1 \bmod 91$
For $a = 16$, $b = a^r = 16^{45} \equiv 1 \bmod 91$
For $a = 75$, $b = a^r = 75^{45} \equiv 1 \bmod 91$
Output: 91 is pseudoprime with the error rate $E_n(t) = \frac{1}{4^3}$

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 $\boldsymbol{a}^{\boldsymbol{r}} \not\equiv \mathbf{1} \bmod \boldsymbol{n}$ and the equation $\boldsymbol{a}^{\mathbf{2}^{\boldsymbol{j}}\boldsymbol{r}} \not\equiv -\mathbf{1} \bmod \boldsymbol{n}$ 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 $\boldsymbol{n}$.

In Example 4, $a = 2$ is a strong witness for the composite number 91. In Example 5, $a = 9, a = 16$, $a = 75$ are strong liars for the composite number 91.

In the following section, we review the RSA algorithm.

3. RSA ALGORITHM

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

The RSA cryptosystem is based on the product of two large prime numbers. The security of RSA relies on the difficulty of factoring a large integer that is the product of two sufficiently large prime numbers. The RSA algorithm's reliability is directly proportional to the size of the prime numbers. The RSA cryptosystem is the most widely used public-key cryptography scheme. The RSA system is used in many application areas such as SSL/TLS protocol, S-MIME, S/WAN, STT and web security certificates for credit card transactions.

The RSA algorithm has three main components: key generation, encryption and decryption. We assume that person Alice wants to send a secret message $\boldsymbol{m}$ to person Bob. Bob generates a key pair: a public key and a private key for the RSA algorithm. Alice encrypts a message $\boldsymbol{m}$ by using Bob's public key.

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

Bob follows the RSA key generation steps.

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 Euler's Totient function $\boldsymbol{\Phi(n) = (p-1)\cdot(q-1)}$ is calculated.
4. A random number $\boldsymbol{e}$ is selected from $\boldsymbol{1 < e < \Phi(n)}$ such that $\boldsymbol{\gcd\left(e, \Phi(n)\right) = 1}$.
5. The value of $\boldsymbol{d}$ is found such that $\boldsymbol{e \cdot d \equiv 1 \bmod \Phi(n)}$.

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

Alice obtains Bob's public key pair $(\boldsymbol{n}, \boldsymbol{e})$ and encrypts a message $\boldsymbol{m}$ as follows.

RSA Encryption

- The message $\boldsymbol{m}$ is written in the range $\boldsymbol{1 \le m \le n-1}$.
- Alice encrypts $\boldsymbol{c \equiv m^e \bmod n}$.
- Alice sends the encrypted message $\boldsymbol{c}$ to Bob.

Bob receives the encrypted message $\boldsymbol{c}$ from Alice and decrypts it by using his private key $\boldsymbol{d}$.

RSA Decryption

- Bob decrypts $\boldsymbol{m \equiv c^d \bmod n}$
- Bob obtains the original message $\boldsymbol{m}$.

Hence, Alice and Bob establish a secure communication using the RSA algorithm. The security of the RSA algorithm derives from the difficulty of factoring large numbers. The public key and private key 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 corresponding 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 the secure prime numbers of p and q such that $n = p\,q$ are resistant to factorization algorithms. Therefore, prime numbers p and q should be selected according to certain criteria [2]. The selected parameters provide a security level that is proportional to the size of the RSA modulo n [12].

The paper [11] discusses how the RSA system can be used in the upcoming era of electronic mail. In the paper [9], the measurement of the distance between the selected primes p and q for RSA is defined. In the book [10], the authors explain the most important techniques of modern cryptography. In the paper [7], the author uses the perfect secure prime number sequence defined in a new method for finding prime numbers in the RSA encryption method. For more details about the RSA system, the reader is directed to the main works [2,5,7,9,10,11].

4. THE PERFORMANCE ANALYSES OF THE PRIMALITY TESTS

In this section, we discuss the performance analyses of the probabilistic primality test algorithms.

We implement in the C++ programming language the probabilistic primality tests such as the Fermat, Solovay-Strassen and Miller-Rabin tests given in Algorithms 1, 2 and 3. This section aims to perform and compare the performance analyses of these tests. The following criteria such as runtime, memory requirements and the number of operations are considered in the analysis of the performance of these tests. When we compare the Fermat, Solovay-Strassen and Miller-Rabin primality tests, we observe that the Miller-Rabin test performs better than the others in terms of error rate and runtime. The error rate of the Fermat test is rather high, and so it is weak in detecting Carmichael numbers. The Solovay-Strassen test has a high runtime due to Jacobi symbol calculations. Additionally, while the Fermat test and the Solovay-Strassen perform with an error rate of $\boldsymbol{E_n}\ (\boldsymbol{t}) = \frac{\mathbf{1}}{\mathbf{2^t}}$, the Miller-Rabin test provides more accurate results with an error rate of $\boldsymbol{E_n}\ (\boldsymbol{t}) = \frac{\mathbf{1}}{\mathbf{4^t}}$ (see in [12] for more detail).

Below, we compare the performance of the probabilistic primality tests 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 presented in Table 1.

Table 1. Fermat Test runtime

FERMAT TEST		
Number of Digits	Mersenne Number	Runtime (seconds)
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 presented in Table 2.

Table 2. Solovay-Strassen Test runtime

SOLOVAY-STRASSEN TEST		
Number of Digits	Mersenne Number	Runtime (seconds)
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 preented in Table 3.

Table 3. Miller-Rabin Test runtime

MILLER-RABIN TEST		
Number of Digits	Mersenne Number	Runtime (seconds)
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, we observe that the Miller-Rabin test is faster than the Solovay-Strassen test. The runtimes of the Miller-Rabin and Solovay-Strassen tests are given in Figure 1.

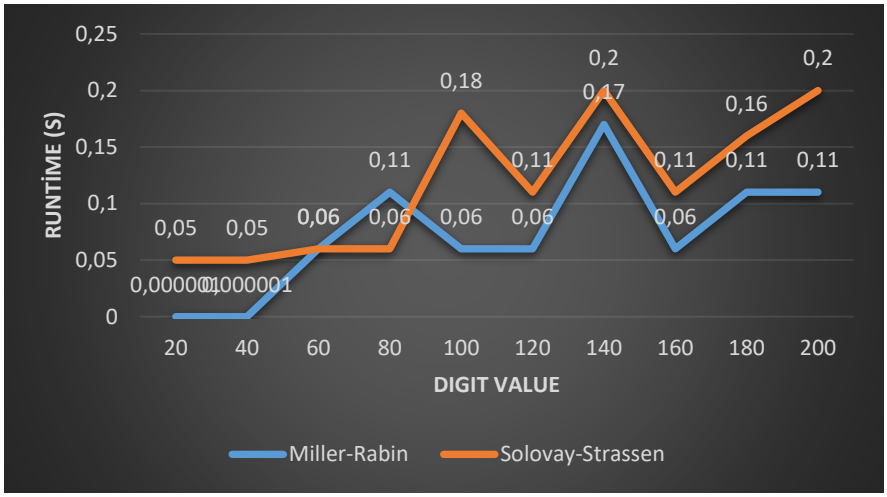

Figure 1. Comparison of the runtimes of the Miller-Rabin and Solovay-Strassen tests

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 required. Thus, finding sufficiently large prime numbers is a significant research problem in the literature. To generate large prime numbers, the probabilistic primality tests are used in cryptography. In this paper, we review the probabilistic primality tests such as the Fermat, Solovay-Strassen and Miller-Rabin test algorithms. Moreover, the performance analyses of the Fermat, Solovay-Strassen and Miller-Rabin algorithms 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 error rate 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*, Citeseer, 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