MATEC Web of Conferences **197**, 03005 (2018) https://doi.org/10.1051/matecconf/201819703005
AASEC 2018

Pseudo-prime number simulation and its application for security purpose

Robbi Rahim[1*], *Nuning* Kurniasih[2], *Nurmaliana* Pohan[3], *S* Sriadhi[4], *Tri* Listyorini[5], *Ricardo Freedom* Nanuru[6], *Rosida Tiurma* Manurung[7], *Asep* Najmurrokhman[8], *Abdurrozzaq* Hasibuan[9], *Dahlan* Abdullah[10], and *Darmawan* Napitupulu[11]

[1]Universiti Malaysia Perlis, School of Computer and Communication Engineering, Kubang Gajah, Malaysia
[2]Universitas Padjadjaran, Faculty of Communication Science, Library and Information Science Program, Bandung, Indonesia
[3]Indonesian Publications Collaboration Community, Indonesia
[4]Universitas Negeri Medan, Department Education of Information Technology and Computers, Medan, Indonesia
[5]Universitas Muria Kudus, Department of Informatics, Kudus, Indonesia
[6]Universitas Halmahera, Maluku Utara, Indonesia
[7]Universitas Kristen Maranatha, Bandung, Indonesia
[8]Universitas Jenderal Achmad Yani, Department of Electrical Engineering, Cimahi, Indonesia
[9]Universitas Islam Sumatera Utara, Department of Industry Engineering, Medan, Indonesia
[10]Universitas Malikussaleh, Department of Informatics, Lhokseumawe, Indonesia
[11]Indonesian Institute of Sciences, Research Center for Quality System and Testing Technology, Jakarta, Indonesia

Abstract. Many public cryptography schemes rely on the use of prime numbers like for encryption and decryption. A prime number is one number that is widely used and large and consists of hundreds of digits, so it takes time to test whether the numbers are prime or not. Miller-Rabin is one algorithm that could be used to test prime number. Simulation to show how to test non-prime number elimination process can be used to determine the workings of the Miller-Rabin algorithm and also could be used as a media learning for students and lecturers to know how prime number test and generation.

1 Introduction

In mathematics, prime numbers are natural numbers that only have two factors, namely the divisor is 1 and the number itself [1]. The uniqueness of prime numbers is widely used in cryptographic algorithms especially on key generation [2–4], the strength of a cryptography algorithm depends on the key used [5–7] and prime number is one of the factors that determine the strength of security [8–10]. Few algorithms using a prime number for the key is RSA, RC4 and Blowfish algorithm using the prime number in p and q key for encryption and decryption process [11–14], this prime numbers become essential not only in mathematics but also in computer science.

The problem that arises is that the prime number is infinite. Not only that, many numbers are glimpsed like primes, but in reality they are not [1,15]. This problem causes the search for prime numbers takes longer time for using as key in cryptography algorithms. An alternative that can be used to overcome this problem is to use the prime test algorithm [15].

One of the prime number tests is Miller-Rabin algorithm. The algorithm works by utilizing the Fermat method where each number tested will go through the process of elimination based on the probability of the prime rate of that number [16]. The output of the Miller-Rabin algorithm is entirely accurate, and the workflow is easy to follow, this reason makes Miller-Rabin algorithm suitable to be implemented in the form of a prime generator software, especially simulation software and also it can be used for security and education purpose.

2 Methodology

2.1 Prime Numbers

The prime number is a positive integer a, where $a \geq 2$ can only be divided by 1 and the number itself. The nature of division of integers gives birth to the concepts of prime numbers and modulo arithmetic [1,15,17–19]. Most public-key algorithms use prime numbers as one of their parameter values [20,21]. There are several important properties that only prime numbers have:

a. All prime numbers are odd numbers, except 2.
b. The number of primes that will not exceed x is $\pi(x)<x$.
c. Hadamard Poussin's theorem states that approach expresses the number of primes for $x \rightarrow \infty$:
$$\pi(x) \approx x / (\ln(x))$$
d. Any positive integer more than one has a prime divisor.
e. If n is a composite number, then it has a prime factor not greater than $\sqrt{n}$.
f. Any integer greater than 1 can be denoted singly as a result of multiplication of prime numbers.

*Corresponding author: usurobbi85@zoho.com

g. If the prime number *p* divides the positive integer n a1, a2, ..., an, then there ai, $1 \leq i \leq n$, such that p divides ai.
h. Two consecutive odd numbers p and p + 2, both of which are prime numbers, are called twin prime pairs. The pair (3.5), (5,7), (11,13), (17,19), (29,31), etc. are the twin prime pairs.

2.2 Miller-Rabin

The Miller-Rabin algorithm is based on Fermat's theorem states that $a^{n-1} \equiv 1 \, mod \, n$ if *n* is a prime and a root or solution *x of* x^2 *(mod n)* having at least four roots if *n* is complex [22,23].

Miller-Rabin algorithm workflow can be described as follows:

a. Take *a* random located at interval 1 <a ≤ n-1 and then count using equation $T = a^m \, mod \, n$ if T = ± 1, then it is concluded *n* may be prime.
b. If $T^2 = 1$, then it is concluded n compound, and the algorithm stops. If $T^2 = -1$, then *n* may be primed. If $T^2 \neq 1$, proceed to step 3.
c. Calculate$T^{2^2} = a^{2^2 X \, m} \, mod \, n$. If $T^{2^2} = 1$ then it is concluded n compound and algorithm will be stopped. If T ^ (2 ^ 2) = -1 then concluded n may be prime. If $T^{2^2} \neq \pm 1$ the process (step 3) is repeated until $T^{2^{k-1} X \, m}$ if until the last iteration is obtained $T^{2^{k-1}} > 1 \, (T \neq \pm 1)$, concluded *n* compound.

3 Results and Discussion

The simulation of the prime generator using Miller-Rabin algorithm is performed by generating and testing randomly generated numbers, and a simple experiment was to count 13 is it prime or not.

a. Calculating the Value of *s* and *d*
Before calculating the value of *s* and *d*, first calculated the value of *N-1*, where *N* is the number to be checked the primes value, from the case example, we get *N* -1 as follows:
N-1 = 13-1
= 12
Next is calculated the value of *s* and *d* by using the formula d * 2^s = N-1, where *d* is a positive odd number, d> 0 and s ≥ 1.
b. Determining the Level of Accuracy
The process of testing prime numbers will do the generation of random numbers that serve as test numbers. The amount of generation of this test number will determine the accuracy of the results of primes
c. Generating Test Numbers
The Miller-Rabin algorithm uses a random number *a*, which will be used to find the value of *x* through the following equation: x = a^d mod N. If the value of x = 1 or the value of x = N-1, then this step will result in a True value for the test number a.
If the value of x does not meet the above requirements, search for the next x value by using the following equation: $x = a^{2^i * d} \, mod \, N$, where *i* is any value from 1 to s-1. If any of the values are x = N-1, then this step will result in a True value for the test number a
d. Testing Prime Numbers
After obtaining the value of *s*, *d* and *a*, the next step is to test the number *N*. In this case, for example, will be tested N = 19 with k = 4 accuracy level used. For k1, we get the test number a = 10. The value of x obtained is:
x = 109 mod19
= 1000000000 mod 19
= 18
Since the value of x = N-1, then the first test is True. For k2, we get the test number a = 18. The value of x obtained is:
x = 189 mod 19
= 198359290368 mod 19
= 18
Since the value of x = N-1, then the second test is True. For k3, the test number obtained a = 17. The value of x obtained is:
x = 179 mod 19
= 118587876497 mod 19
= 1
Since the value of x = 1, then the third test is True. For k4, the test number a = 9. The x values obtained are:
x = 99 mod 19
= 387420489 mod 19
= 1
Since the value of x = 1, then the third test is True. Since all tests from k1 to k4 are true, it can be concluded that number 19 is a prime number.

The Miller-Rabin algorithm simulation created using Java Netbeans displays the result of prime numbers of n with varying amount of time, the designed app displays the numbers that are prime numbers and the required processing time as follows:

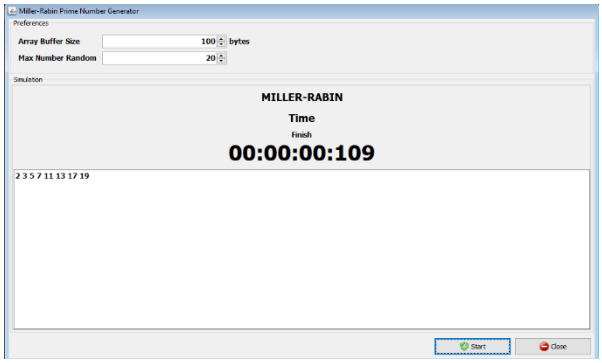

Fig.1. Application Simulation.

Figure 1 is a simple example of displaying a prime number of *n*, the number of *n* is randomly generated and then tested by using the prime algorithm individually stored in the array and then displayed the result, some other tests as in Figure 2-5 with the value large enough with varying process time.

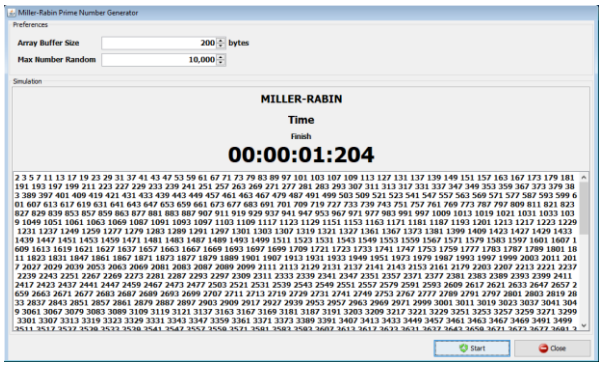

Fig.2. A sample of 10000 number.

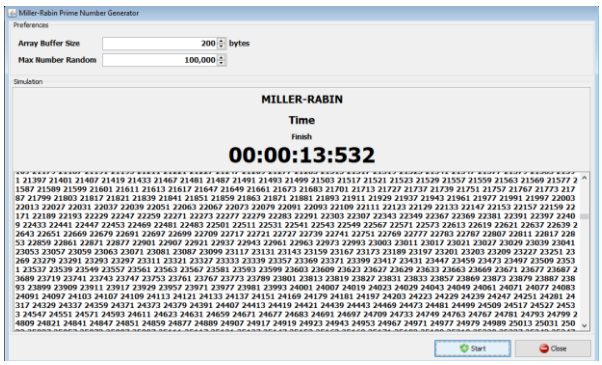

Fig.3. A sample of 100000 number.

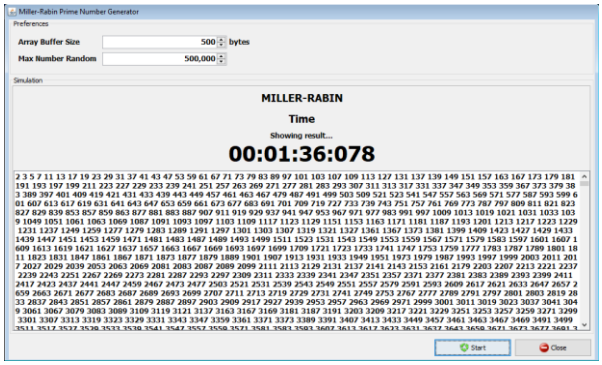

Fig.4. A sample of 500000 number.

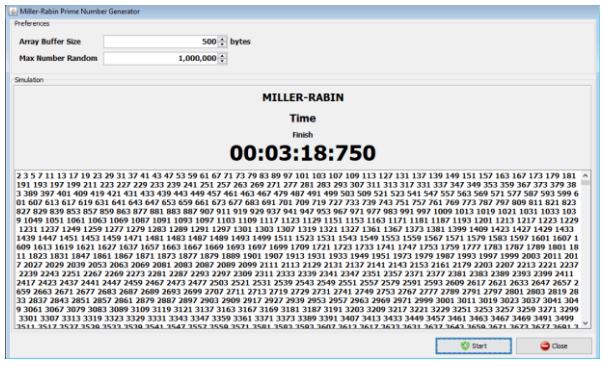

Fig.5. A sample of 1000000 number.

The experiment shows the number of primes as much as *n* number, based on the test that has been done the more prime numbers generated and tested using Miller-Rabin algorithm then it takes a relatively long time, one example is to generate and test as many as 100000 prime numbers require time 13 second but for 1000000 primes takes 3 minutes, for some experiment can be seen in table 1 below:

Table 1. Prime Number Experiment.

No	Prime Number	Miller-Rabin (Second)
1	10	0:0:0.109
2	10000	0:0:1.204
3	100000	0:0:13.532
4	500000	0:1:36.078
5	1000000	0:3:18.750

Based on table 1 it will like this figure 6 below where the process will take exponential in time.

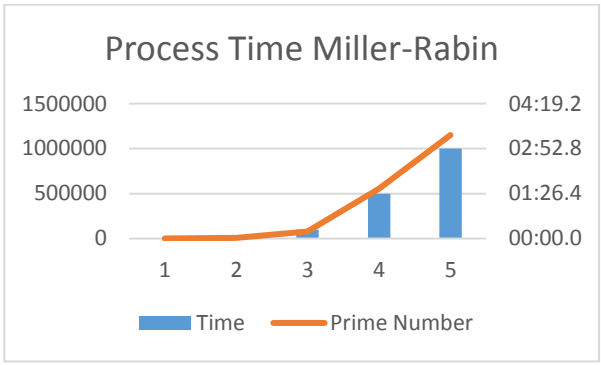

Fig.6. Graph Miller-Rabin process.

The prime numbers generated in the application are numerous, whereas the use of *p* and *q* keys in cryptography requires only 2 pieces and must use large number, assuming the key value of *p* and *q* above 500000 <10000000 then the required time will not be up to 3 minutes to get value of *p* and *q*.

4 Conclusion

Testing prime numbers using the Miller-Rabin algorithm can be done well and the results are also entirely accurate even up to 1 million prime numbers can be completed by 3 minutes, the Miller-Rabin algorithm is very appropriate as an additional algorithm in the cryptographic process so that the determination of the key p and q which are generally prime numbers in cryptography like RSA, RC5, Pohlig-Hellman can be generated quickly.

References

1. D. Abdullah, R. Rahim, D. Apdilah, S. Efendi, T. Tulus, and S. Suwilo, "Prime Numbers Comparison using Sieve of Eratosthenes and Sieve of Sundaram Algorithm," in Journal of Physics: Conference Series, vol. **978**, no. 1, p. 012123. (2018)
2. S. Bruce, *Applied cryptography*. 1996.
3. M. Blumenthal, "Encryption: Strengths and Weaknesses of Public-key Cryptography," *CSRS 2007*, pp. 1–7, (2007)
4. R. Rahim, "Man-in-the-middle-attack prevention using interlock protocol method," ARPN J. Eng. Appl. Sci., vol. **12**, no. 22, pp. 6483–6487, (2017)

5. D. Schmidt and T. Jaeger, "Pitfalls in the automated strengthening of passwords," in *Proceedings of the 29th Annual Computer Security Applications Conference on - ACSAC '13*, pp. 129–138. (2013)
6. H. Nurdiyanto, R. Rahim, and N. Wulan, "Symmetric Stream Cipher using Triple Transposition Key Method and Base64 Algorithm for Security Improvement," J. Phys. Conf. Ser., vol. **930**, no. 1, p. 012005, Dec. (2017)
7. R. Rahim *et al.*, "Combination Base64 Algorithm and EOF Technique for Steganography," J. Phys. Conf. Ser., vol. **1007**, no. 1, p. 012003, Apr. (2018)
8. H. K. Sahu, V. Jadhav, S. Sonavane, and R. K. Sharma, "Cryptanalytic attacks on international data encryption algorithm block cipher," Def. Sci. J., vol. **66**, no. 6, pp. 582–589, (2016)
9. D. Estes, L. M. Adleman, K. Kompella, K. S. McCurley, and G. L. Miller, "Breaking the Ong-Schnorr-Shamir Signature Scheme for Quadratic Number Fields BT - Advances in Cryptology — CRYPTO '85 Proceedings," H. C. Williams, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 3–13. (1986)
10. R. Rahim, N. Kurniasih, M. Mustamam, L. Andriany, U. Nasution, and A. H. Mu-, "Combination Vigenere Cipher and One Time Pad for Data Security," *Int. J. Eng. Technol.*, vol. **7**, no. 2.3, pp. 92–94, (2018)
11. D. Chandravathi and P. V. Lakshmi, "Advanced Homomorphic Encryption for Cloud Data Security," JOIV Int. J. Informatics Vis., vol. **1**, no. 4, p. 1, Mar. (2017)
12. B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition," *Network*. pp. 623–631, (1996)
13. E. Kartikadarma, T. Listyorini, and R. Rahim, "An Android mobile RC4 simulation for education," World Trans. Eng. Technol. Educ., vol. **16**, no. 1, pp. 75–79, (2018)
14. R. Rahim, M. Dahria, M. Syahril, and B. Anwar, "Combination of the Blowfish and Lempel-Ziv-Welch algorithms for text compression," World Trans. Eng. Technol. Educ., vol. **15**, no. 3, pp. 292–297, (2017)
15. R. Rahim, H. Winata, I. Zulkarnain, and H. Jaya, "Prime Number: an Experiment Rabin-Miller and Fast Exponentiation," J. Phys. Conf. Ser., vol. **930**, no. 1, p. 012032, Dec. (2017)
16. P. Brodanac, L. Budin, and D. Jakobovic, "Parallelized Rabin-Karp method for exact string matching," *Proc. ITI 2011, 33rd Int. Conf. Inf. Technol. Interfaces*, pp. 585–590, (2011)
17. R. Rahim and A. Ikhwan, "Study of Three Pass Protocol on Data Security," *Int. J. Sci. Res.*, vol. **5**, no. 11, pp. 102–104, Nov. (2016)
18. D. Nofriansyah *et al.*, "A New Image Encryption Technique Combining Hill Cipher Method , Morse Code and Least Significant Bit Algorithm," J. Phys. Conf. Ser., vol. **954**, no. 1, p. 012003, (2018)
19. H. Nurdiyanto and R. Rahim, "Enhanced pixel value differencing steganography with government standard algorithm," in *2017 3rd International Conference on Science in Information Technology (ICSITech)*, pp. 366–371. (2017)
20. R. Rahim, "128 Bit Hash of Variable Length in Short Message Service Security," Int. J. Secur. Its Appl., vol. **11**, no. 1, pp. 45–58, Jan. (2017)
21. R. Rahim and A. Ikhwan, "Cryptography Technique with Modular Multiplication Block Cipher and Playfair Cipher," Int. J. Sci. Res. Sci. Technol., vol. **2**, no. 6, pp. 71–78, (2016)
22. G. Böckle, "The Miller-Rabin test with randomized exponents," *J. Math. Cryptol.*, vol. **3**, no. 4, pp. 307–319, (2009)
23. J. Hurd, "Verification of the Miller-Rabin probabilistic primality test," J. Log. Algebr. Program., vol. **56**, no. 1–2 SPEC., pp. 3–21, (2003)