Prime Number: an Experiment Rabin-Miller and Fast Exponentiation

To cite this article: Robbi Rahim et al 2017 J. Phys.: Conf. Ser. 930 012032

View the article online for updates and enhancements.

Related content
- On the Hilbert-Kamke problem in prime numbers
  D A Mit'kin
- Path integral approach to eikonal and next-to-eikonal exponentiation
  Eric Laenen, Gerben Stavenga and Chris D. White
- On non-singlet physical evolution kernels and large-x coefficient functions in perturbative QCD
  S. Moch and A. Vogt
Prime Number: an Experiment Rabin-Miller and Fast Exponentiation

Robbi Rahim1*, Hendryan Winata2, Iskandar Zulkarnain2 and Hendra Jaya3

1Postgraduate Ph.D. Student Universiti Malaysia Perlis, Universiti Malaysia Perlis, Malaysia
2Department of Health Information, Akademi Perekam Medik dan Infokes Imelda, Jl. Bilal Ujung Medan 20116, Indonesia
3Department of Computer System, STMIK Triguna Dharma, Jl. Jenderal Abdul Haris Nasution No.73, Medan 20219, Indonesia

*usurobbi85@zoho.com

Abstract. Prime number is one of the most widely used names for various security purposes, especially in the process of encryption and decryption for asymmetric algorithms, the speed in determining prime number becomes one of the most important things, Rabin-Miller algorithm is used to test a number whether including prime or not, speed in determining prime number is very important and technique that could be using is fast exponentiation, this research get significant result in testing prime number with length of variation number.

1. Introduction
Public key cryptography has several issues such as factorization, Knapsack problems, elliptical curves, discrete algorithms, and so on. Factorization [1] [2] [3] [4], significant number modulo powers and prime numbers is a widely used mathematical function in public key cryptography [5] [6]. The process of factorization, large number modulo force and the formation and testing of prime numbers using conventional way often experience obstacles [7] [8], for example in determining the top value of $2^{2000}$ mod $5^{300}$, to know the process of identifying prime numbers can be used algorithms such as Rabin-Miller and Fast Exponentiation, the use of algorithms using the Pseudo Code .Net Language is expected to allow users to design applications prime generator.

2. Theory
A. Basic Mathematical Concepts of Cryptography
The underlying mathematical basis of the process of encryption and decryption is the relation between two sets i.e. the set contains the plaintext element, and the set includes the ciphertext element [2]. Encryption and decryption is a transformation function between the two sets. If the plaintext set is indicated by P and the set of ciphertext is indicated by C, while E denotes the encryption function and
the decryption function with $D$ [2] [8] [9] the encryption-decryption process can be expressed in mathematical notation by:

$E(P) = C$ and

$D(C) = P$

Since the encryption-decryption process aims to recover the original data, then:

$D(E(P)) = P$

The relation between the plaintext set and the ciphertext set must be a one to one relation to preventing the occurrence of ambiguity in decryption i.e. one ciphertext element declares more than one plaintext element.

B. Factorization of Large Numbers

Number theory is the basis of the number itself, and the core blocks of numbers are primes so that primes are often said to be central to number theory [10]. A prime number is a number that only has two factors: the number one and the number itself [10] [11]. Numbers that have more than two elements called composite numbers. Numbers one has only one factor and is considered not prime and composite numbers [10] [12]. When a composite number is composed of a product of all prime factors, it is said to be the main factorization of the number. Factorization of a number into its primary constituent element also called prime decomposition. Typically, factors are listed from the smallest to the largest [9] [10].

C. Rabin-Miller

The simple algorithm used by everyone is designed by Michael Rabin based on some ideas from Gary Miller. The test algorithm is as follows:

1) Select a random number $p$ to test.
2) Calculate $b$, where $b$ is the number $(p - 1)$ divided by 2 (i.e., $b$ is the largest power of 2, such that $2b$ is a factor of $p - 1$).
3) Then calculate $m$, such that $p = 1 + 2^b \cdot m$
4) Choose a random number $a$ such that $a$ is smaller than $p$.
5) Set $j = 0$ and set $z = a^m \mod p$.
6) If $z = 1$ or if $z = p - 1$, then $p$ passes the test and may be a prime number.
7) If $j > 0$ and $z = 1$, then $p$ is not a prime number.
8) Set $j = j + 1$. If $j < b$ and $z \neq p - 1$, set $z = z^2 \mod p$ and return to stage 4. If $z = p - 1$, then $p$ passes the test and may be prime.
9) If $j = b$ and $z \neq p - 1$, then $p$ is not a prime number.

3. Result and Discussion

For testing module velocity with Fast Exponentiation can be taken case example $1123829 \mod 95317$ with the process as follows:

$A1 = 11$

$B1 = 23829$

Product = 1

While $23829 <> 0 \rightarrow True$

While $23829 \mod 2 = 0 \rightarrow False$

$B1 = 23829 - 1 = 23828$

Product = $(1 * 11) \mod 95317 = 11$

While $23828 <> 0 \rightarrow True$

While $23828 \mod 2 = 0 \rightarrow True$
B1 = 23828 div 2 = 11914
A1 = (11 * 11) mod 2 = 121

While 11914 mod 2 = 0  →  True
B1 = 11914 div 2 = 5957
A1 = (121 * 121) mod 2 = 14641

While 5957 mod 2 = 0  →  False
B1 = 5957 - 1 = 5956

While 4 <> 0  →  True
  
  While 4 mod 2 = 0  →  True
  B1 = 4 div 2 = 2
  A1 = (40123 * 40123) mod 2 = 46316

  While 2 mod 2 = 0  →  True
  B1 = 2 div 2 = 1
  A1 = (46316 * 46316) mod 2 = 62771

  While 1 mod 2 = 0  →  False
B1 = 1 - 1 = 0
Product = (11004 * 62771) mod 95317 = 65102

While 0 <> 0  →  False

FastExp(11, 23829, 95317) = 65102

The core of the prime random generation process is to test whether the random number generated is either prime or not. In order to better understand the process of generating prime random numbers, taken for example number 95317 with A = 11, then the process of completion with Rabin Miller algorithm is as follows:

C = 24889 - 1 = 24888
nTemp = 0
24888 mod (2^0) = 0 And ((2^0) < 24889) - True
nTemp = 0 + 1 = 1
24888 mod (2^1) = 0 And ((2^1) < 24889) - True
nTemp = 1 + 1 = 2
24888 mod (2^2) = 0 And ((2^2) < 24889) - True
nTemp = 2 + 1 = 3
24888 mod (2^3) = 0 And ((2^3) < 24889) - True
nTemp = 3 + 1 = 4
24888 mod (2^4) = 0 And ((2^4) < 24889) - False
B = 4 - 1 = 3
M = 24888 / (2^3) = 3111
J = 0
### Table 1. Running Time Execution

| No | Length Number in digit (Random) | With Fast Exponentiation (second) | Without Fast Exponentiation (second) |
|----|--------------------------------|----------------------------------|-------------------------------------|
| 1  | 2                              | 0.09                             | 0.08                                |
| 2  | 3                              | 0.247                            | 0.161                               |
| 3  | 5                              | 1.45                             | 3.14                                |
| 4  | 7                              | 11.39                            | 19.43                               |
| 5  | 9                              | 22.51                            | 35.19                               |
| 6  | 13                             | 73.02                            | 108.49                              |
| 7  | 17                             | 165.58                           | 301.11                              |

Testing performed on a different number to determine prime numbers using fast exponentiation technique known faster if for large numbers compared without using fast exponentiation, for more details could see in the graph below following the resulting time difference.

![Time Execution Prime Number](image)

**Figure 1. Time Execution Prime Number**

The above graph shows the difference in execution time by using Rabin-Miller with fast exponentiation faster to determine prime numbers.

4. Conclusion

The prime Rabin-Miller and Fast Exponentiation test can be using for excellent test of randomly generated random numbers, and by using Rabin-Miller and Fast Exponentiation, algorithms can perform a large test process of up to 15 digits.

References

[1] R. Rahim, "128 Bit Hash of Variable Length in Short Message Service Security," *International Journal of Security and Its Applications*, vol. 11, no. 1, pp. 45-58, 2017.

[2] R. Rahim and A. Ikhwan, "Cryptography Technique with Modular Multiplication Block Cipher and Playfair Cipher," *IJSRST*, vol. II, no. 6, pp. 71-78, 2016.
[3] R. Rahim and A. Ikhwan, "Study of Three-Pass Protocol on Data Security," *International Journal of Science and Research (IJSR)*, vol. 5, no. 11, pp. 102-104, 2016.

[4] A. P. U. Siahaan and R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm," *International Journal of Security and its Applications*, vol. 10, no. 8, pp. 173-180, 2016.

[5] D. Cries and J. Misra, "A Linear Sieve Algorithm for Finding Prime Numbers," *Communication of the ACM*, vol. 21, no. 12, pp. 999-1003, 1978.

[6] Y. Fu and S. Xu, "Generalized Fast Coding Steganographic Algorithm Based on Prime Number," in *IEEE International Conference on Information Theory and Information Security (ICITIS)*, Beijing, China, 2010.

[7] G. A. Jullien, "Implementation of Multiplication, Modulo a Prime Number, with Applications to Number Theoretic Transforms," *IEEE Transactions on Computers*, vol. 29, no. 10, pp. 899-905, 1980.

[8] Y.-S. Kim, S. R. Shrestha, and J.-W. Jang, "Refined Algorithm for Prime Number Generation in Embedded Security Systems," in *IEEE Asia-Pacific Services Computing Conference (APSCC)*, Jeju Island, South Korea, 2011.

[9] V. S. Igumnov, "Generation of the Large Random Prime Numbers," in *5th International Siberian Workshop on Electron Devices and Materials, 2004. Proceedings. 5th Annual. 2004 International Siberian Workshop on*, Erlagol, Altai, Russia, Russia, 2004.

[10] D. Liu, "Iteration Distribution of Prime Numbers," *Journal of Mathematics Research*, vol. 3, no. 4, pp. 112-117, 2011.

[11] Z. Qing and H. Zhihua, "The large prime numbers generation of RSA algorithm based on genetic algorithm," in *International Conference on Intelligence Science and Information Engineering*, Wuhan, China, 2011.

[12] R. R. Sarukkai, "Prime Numbers and Output Codes," in *IEEE International Conference on Neural Networks*, Perth, WA, Australia, 1995.