The IJICS (International Journal of Informatics and Computer Science)
Vol 4 No 2, September 2020

ISSN 2548-8384 (online)
ISSN 2548-8449 (print)
Page 35 – 38
DOI 10.30865/ijics.v4i2.1386

Middle Square Method Analysis of Number Pseudorandom Process

Arif Budiman, Efori Buulolo, Imam Saputra

Department of Computer Science, Universitas Budi Darma, North Sumatra, Indonesia
Email: arif@gmail.com

Abstract−Random numbers can be generated from a calculation of mathematical formulas. Such random numbers are often referred to as pseudo random numbers, random numbers are used for various algorithms, especially cryptographic algorithms such as AES, RSA, IDEA, GOST that require the use of Middle-Square Method random numbers which is very useful for adding research references to algorithms concerning random number generator and better understand how random numbers are generated using the Middle-Square Method algorithm. Both data collection and report making as for the objectives achieved in the form of understanding random numbers and knowing the algorithm process, compile a program designed to be used as an alternative to random numbers for various purposes, especially in cryptographic algorithms.

Keywords: Middle Square, Pseudorandom Number

1. INTRODUCTION

Randomness is an uncertain / unpredictable event, and the attribute that makes randomness is random numbers. Randomness can be generated by environmental conditions, randomness from initial conditions (chaos theory), and random numbers generated by the system. The system generates "random" numbers with certain computational algorithms. The algorithm is called a pseudo random number generator.

Random numbers can be generated from a calculation of mathematical formulas. Such random numbers are often referred to as pseudo random numbers [1], random numbers are used for various algorithms, especially cryptographic algorithms such as AES, RSA, IDEA, GOST which require the use of random numbers.

Random numbers in cryptographic algorithms in particular can not be avoided, one of the factors of this random number is the relationship between random values generated by other random values, prime numbers is one number that uses random numbers as the basis for generating values [2], Blum-Blum Shub , Blum-Micali, Complementary-multiply-with-carry, Linear feedback shift register and Middle-Square Method are random number generator algorithms, Middle-Square Method algorithm that writers use in this thesis research.

This Middle-Square Method was discovered by John von Neumann and Metropolis [3]. The main process of this algorithm selects positive integers called (Z0) in the form of positive integers, then (Z0) is squared to form a digit twice the number of digits Z0, this process is carried out up to n random numbers.

Research related to the development or discussion of the Middle-Square Method algorithm, rahimov [3] improvised the Middle-Square Method algorithm by generating random numbers based on the depth of an RGB image so that a change of 1 bit pixel will change all the values of random numbers generated, this process takes longer compared to the original algorithm but the resulting value reaches 512 bit numbers.

The use of random numbers is not without reason, many algorithms require random numbers as a process such as cryptography and steganography, random number generation may be input manually but by using a combination algorithm each number cannot be predicted so that the numbers become truly random.

Random number generator is needed, especially on various cryptographic algorithms, for example, the AES and RSA algorithm which consists of many numbers for generating existing values, such as prime numbers used in the RSA algorithm for key processes, encryption and decryption, very small prime numbers used the manual input for it must be generated randomly using a random number generator algorithm, one of which is the Middle-Square Method algorithm.

2. THEORY

2.1 Middle-Square Method

Middle-Square Method is a method that produces pseudo random numbers, in practice algorithms that generate random numbers are all not a good algorithm [1]. This is because the process of generating fast and random numbers are positive numbers, but despite having a number weaknesses Pseudo random generated random number generator algorithms such as Middle-Square Method is still widely used in cryptographic algorithms because random numbers can not necessarily be generated [5].

ISSN 2548-8384 (online)
ISSN 2548-8449 (print)
Page 35 – 38
DOI 10.30865/ijics.v4i2.1386

This Middle-Square Method was created by John von Neumann to produce a sequence of 4-digit pseudorandom numbers, the initial 4-digit value is made with squares to produce an 8-digit number. If the random number is less than 8 digits, then a value of 0 is given to complete the 8-digit [1]
For the n-digit number generator, the spin process is no more than 8n, if the resulting value is 4 digits then the value of 0 may be added, here are the Middle-Square Method process steps.

1. Choose positive integers as seed (Z0) in the form of positive integers consisting of several digits.
2. Squaring the number so that it forms a digit twice the number of digits Z0, if not, add digit 0 in front of the number.
3. Take the number of digits that correspond to the number of digits Z0 in the middle to be Z1.
4. Add decimal digits in front of Z1.
5. Do it until step n.

2.2. Pseudorandom

Random numbers are numbers that cannot be predicted to occur and basically random numbers can be generated using mathematical formulas with certain patterns that follow certain distribution models [1].

3. RESULT AND DISCUSSION

The process of generating random numbers is a process that is widely used in various algorithms, especially cryptographic and sorting algorithms, with the application of Middle-Square Method the random numbers generated will be more random by not having a pattern that has the same similarity.

3.1 *Analysis of the Middle-Square Method Process*

The process of generating random numbers with the middle square method is done by determining the values of p and q which are prime numbers from which the values of p and q are obtained by the coefficient of n, for example the test can be seen below.
Determined the value of p = 11351 and q = 11987 so that n = pq = 136064437. Then determined.
The value of s = 80331757 and j = 4 (j does not exceed log2 136064437) = 4.75594).
We count x0 = 803317572 mod 136064437 = 131273718.
The random bit sequence we produce is as follows:

$x_1 = x_0^2 \ mod \ n = \ 131273718^2 \ mod \ 136064437 = 47497112$, sehingga $z_1 = 47497112 \equiv 8 \ mod \ 2^4$
$= 1000_{basis2}$
$x_2 = x_1^2 \ mod \ n = \ 47497112^2 \ mod \ 136064437 = 69993144$, sehingga $z_2 = 69993144 \equiv 8 \ mod \ 2^4$
$= 1000_{basis2}$
$x_3 = x_2^2 \ mod \ n = \ 69993144^2 \ mod \ 136064437 = 13810821$, sehingga $z_3 = 13810821 \equiv 5 \ mod \ 2^4$
$= 0101_{basis2}$

X1, X2 and X3 are variables used to get the calculation process values from the modulation process p, q and n and to facilitate the process made using base 2 so that the sequence of random bit blocks produced is 1000 1000 0101 or in decimal is: 8 8 5 , the number is a random number generated by the Middle-Square Method algorithm. To facilitate the manual calculation process the authors use a process like the following:
p = 290033
q = 159503
n = 175807
X1 = 84
X1 = 84 ^ 159503 mod 290033 = 71382
71382 to byte array = [214][22][1][0]
X2 = 101
X2 = 101 ^ 159503 mod 290033 = 203074
203074 to byte array = [66][25][3][0]
X3 = 115
X3 = 115 ^ 159503 mod 290033 = 180307
180307 to byte array = [83][192][2][0]
X4 = 116
X4 = 116 ^ 159503 mod 290033 = 256888
256888 to byte array = [120][235][3][0]
X5 = 105
X5 = 105 ^ 159503 mod 290033 = 142517
142517 to byte array = [181][44][2][0]

ISSN 2548-8384 (online)
ISSN 2548-8449 (print)
Page 35 – 38
DOI 10.30865/ijics.v4i2.1386

X6 = 110
X6 = 110 ^ 159503 mod 290033 = 105820
105820 to byte array = [92][157][1][0]
X7 = 103
X7 = 103 ^ 159503 mod 290033 = 255434
255434 to byte array = [202][229][3][0]
X8= 32
X8 = 32 ^ 159503 mod 290033 = 32530
32530 to byte array = [18][127][0][0]
X9 = 80
X9 = 80 ^ 159503 mod 290033 = 166784
166784 to byte array = [128][139][2][0]
X10 = 101
X10 = 101 ^ 159503 mod 290033 = 203074
203074 to byte array = [66][25][3][0].

4. IMPLEMENTATION

Program testing is used to see the results that have been discussed in the previous chapter, here are the commands for generating random numbers with applications designed using the Dev C ++ programming language

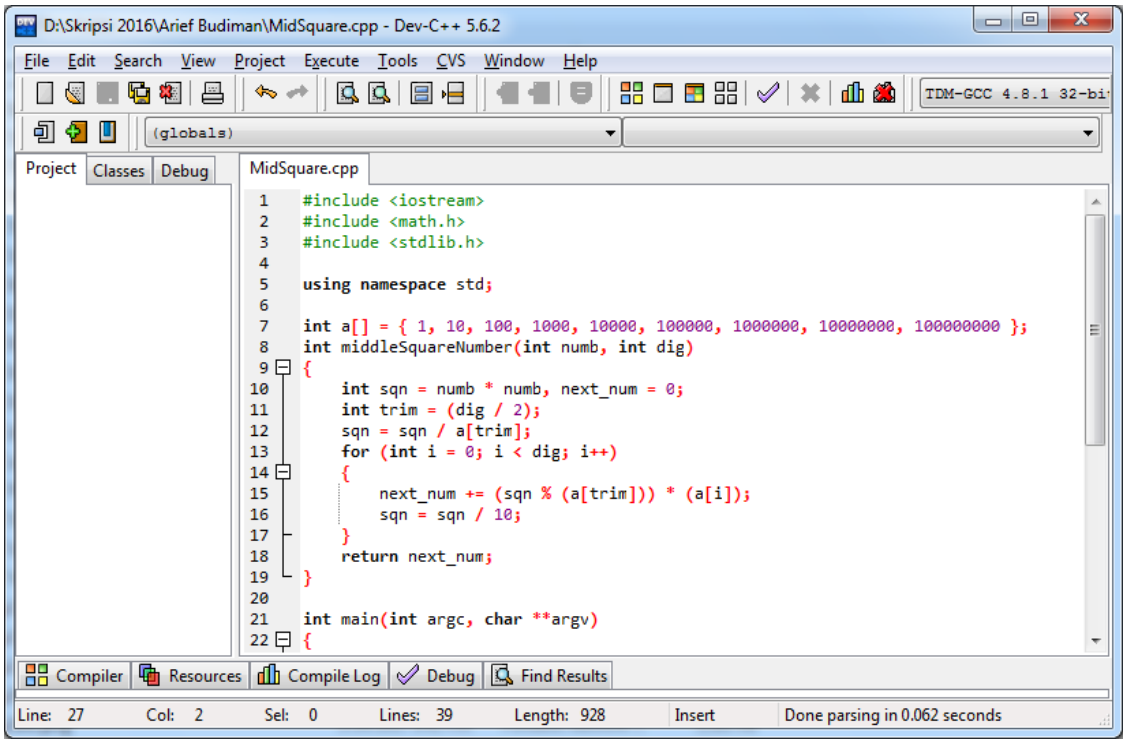

Figure 1. Middle Square Command

After the command has finished typing, the next is to run the program by pressing the F9 key and the results are as follows:

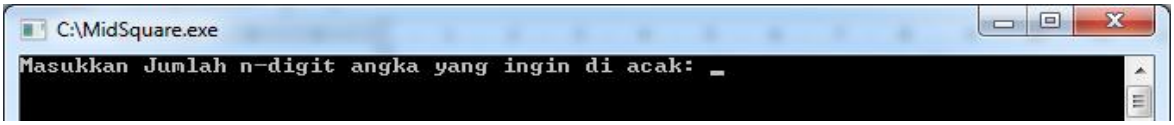

Figure 2. Program Results

After the program is run next is to determine the digits to be randomized, for example the author enters 3 digits and the results are as follows:

The IJICS (International Journal of Informatics and Computer Science)
Vol 4 No 2, September 2020
ISSN 2548-8384 (online)
ISSN 2548-8449 (print)
Page 35 – 38
DOI 10.30865/ijics.v4i2.1386

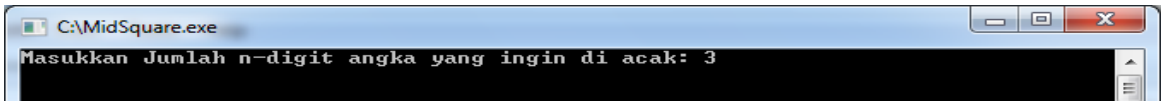

Figure 3. Randomization of 3 digits

After entering the number of digits you want to randomize, next press the enter button and the results are as follows:

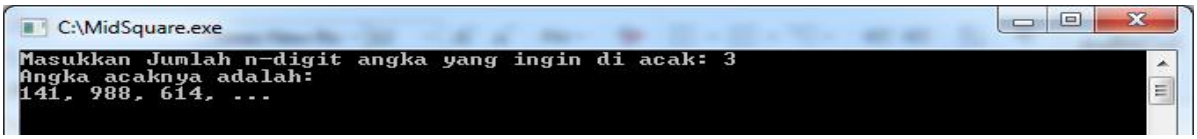

Figure 4. 3-digit randomization results

The next test the author tries to do the test with 9 digits and the results obtained are as follows:

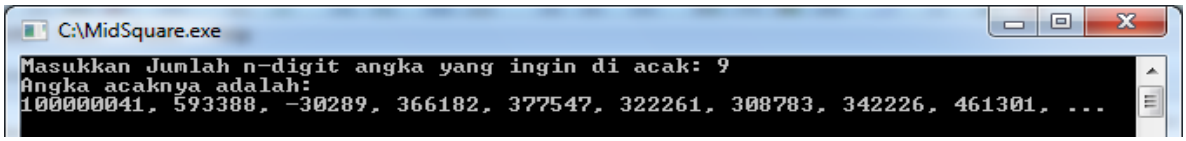

Figure 5. 9 digit randomization results

From the process of testing the program, it was obtained that the random numbers generated were really very random and also fast.

5. CONCLUSION

With the completion of this study which includes the use of the Middle Square Methods algorithm in randomization, several conclusions can be obtained:

1. The pseudo random generator application designed is capable of displaying random numbers with good randomness levels.
2. The system is designed by applying the Middle Square Method algorithm to generate random numbers with n digits less than 2 seconds with large numbers up to 1,000,000,000.
3. The application supports positive and negative numbers for the resulting numbers.

REFERENCES

[1] Yosua P.W Simare mare ,"Middle Square Method ", Penerbit Andi, Jogyakarta, Edisi II, 2013.
[2] Primananda Arif Aditya," Dasar-dasar Pemrograman Database Destop dengan Visual Net C ++ 2008", Penerbit PT. Elex MediA Komputino, Jakarta 2013.
[3] Wikipedia, The Free Encyclopedia [Online],Available :http://en.wikipedia.org John von Neuman.
[4] Abdul Kadir (2003 : 12)
[5] (sumber:http://thesis.binus.ac.id/asli/Lampiran/2006-2-00821-KA Lampiran.pdf/09/06)
[6] (Sumber:http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/symbols.html/09/06).