World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 4.326

PERFORMANCE ANALYSIS OF MULTI-LEVEL ALGORITHM FOR DATA STORAGE SECURITY IN CLOUD COMPUTING

Ismail Ahmed*[1] and Rashid Husain[2]

[1]Research Scholar, Department of Mathematics and Computer Science, UMYU, Katsina.

[2]Lecturer, Department of Mathematics and Computer Science, UMYU, Katsina.

Article Received on 29/07/2017 **Article Revised on 19/08/2017** **Article Accepted on 09/09/2017**

***Corresponding Author**
Ismail Ahmed
Research Scholar, Department of Mathematics and Computer Science, UMYU, Katsina.

ABSTRACT

Internet is a public-interacted system in which the amount of information exchanged over the Internet is completely not safe. Protecting the information transmitted over the network is a difficult task and the data security issues become increasingly important. At present, various types of cryptographic algorithms provide high security of data on the Internet, but still there are also some drawbacks. To enhance the strength of these algorithms, designed a Multi-level Algorithm. The algorithm designed using combination of two asymmetric cryptographic algorithms these two algorithms are Multi-prime RSA and MD5 algorithm. The performance of Multi-level algorithm analyzed by comparing it with multi-prime RSA and MD5 cryptographic algorithms in terms of execution time and memory space.

KEYWORDS: Cryptography, Symmetric, Asymmetric, Multi-Prime RSA, and MD5.

INTRODUCTION

Cloud computing can be considered a new computing paradigm that allows users to temporary utilize computing infrastructure over the network, supplied as a service by the cloud-provider at possibly one or more levels of abstraction (Rao, *et. al.*, 2009). The word cryptography comes from the two Greek words – Crypto meaning secret or hidden and graph meaning writing. It is the practice and study of hiding information. The term refers to the science and art of transforming messages to make them secure and immune to attacks. The

schemes used for encryption constitute this area (Forouzan, 2017). There are a number of *cryptographic primitives*— basic building blocks, such as *block ciphers, stream ciphers*, and *hash functions.* Block ciphers may either have one key for both encryption and decryption, in which case they're called *shared key* (also *secret key* or *symmetric*), or have separate keys for encryption and decryption, in which case they're called *public key* or *asymmetric* (Caesar and Kennedy, 2017).

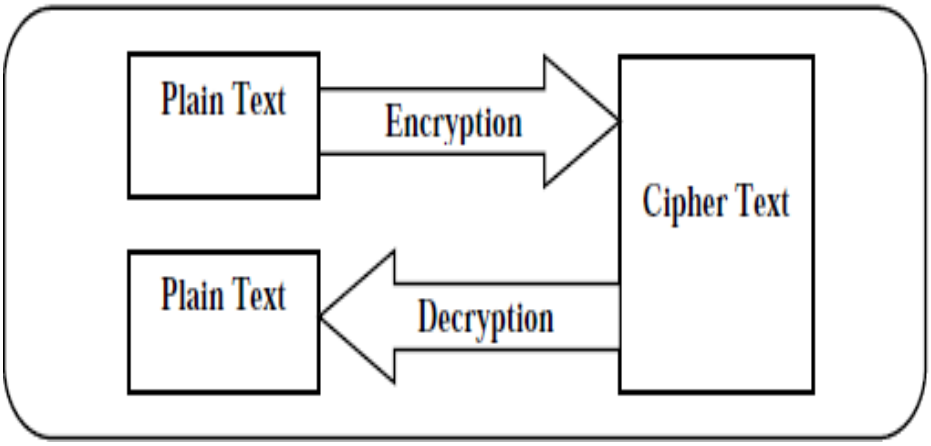

Fig. 1: Simple Cryptosystem.

Categories of Cryptography

The cryptographic algorithms are divided into two groups: symmetric key cryptography algorithms and asymmetric key cryptography algorithms.

Symmetric Key Cryptography

In a symmetric key cryptography, the same key is used by both the parties. The sender uses this key and encryption algorithm to encrypt the data; the receiver uses the same key and the corresponding decryption algorithm to decrypt the data. The key is shared. It is also known as secret-key cryptography *asymmetric* (Caesar and Kennedy, 2017). Symmetric algorithms have the advantage of not consuming too much of computing power and it works with high speed in encrypting them (Triphathi and Agrawal, 2014). Some of the currently used cryptographic technologies in symmetric key cryptography are DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm), Blowfish, RC5 (Rivest Cipher 5), AES (Advanced Encryption Standard), ECC (Elliptic Curve Cryptography) etc (Forouzan, 2017).

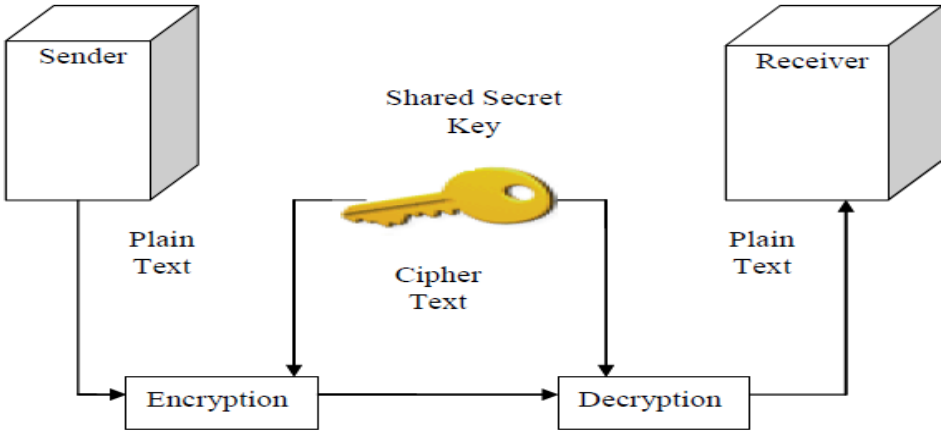

Fig. 2: Symmetric Key Cryptography (Zaren 2011).

Asymmetric Key Cryptography

In asymmetric or public key cryptography, there are two keys: a private key and a public key. The private key is kept by the receiver. The public key is announced to the public. It is also known as public-key cryptography (Forouzan, 2017). Asymmetric key encryption is the technique, in which the different keys are for the encryption and the decryption process. One key is public (published) and second is kept private. The keys used in public-key encryption algorithms are usually much longer that improves the security of the data being transmitted (Triphathi and Agrawal, 2014). Some of the cryptographic algorithms used in asymmetric key cryptography are RSA (Rivest, Shamir, adleman), DH (Diffie-Hellman Key Arrangement Algorithm), ECDH (Elliptic Curve Diffie-Hellman Key Arrangement Algorithm), RPK (Raike Public Key), ElGamal, IES (Integrated Encryption Scheme), CEILIDIH, MD (Message Disgest Algorithms) etc (Triphathi and Agrawal, 2014).

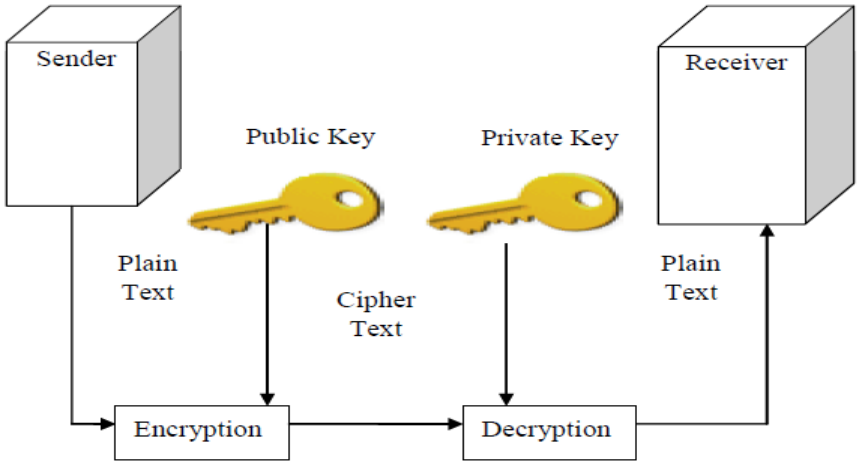

Fig. 3: Asymmetric Key Cryptography (Zaren, 2011).

Multi-Level Cryptographic Algorithm (Ahmed and Husain, 2017)

The Hybrid Cryptographic Algorithm was designed using the combination of two asymmetric key cryptography algorithms namely; Multi-prime RSA which is among the variants of RSA algorithm and MD5 which is also among the variants of Message Digest Algorithms. The designed algorithm consisted of five steps. However, the algorithm is as follows:

Step 1: Key Generation

$For\ any\ integer\ r\ > 2\ (r\ is\ a\ prime\ number)$

$Let\ N\ be\ the\ product\ of\ r, the\ randomly\ chosen\ distinct\ prime\ numbers\ p_1, p_2 \dots p_r$

$Compute\ Euler's\ totient\ function\ of\ N, \phi(N)$

$\phi(N) = (p_1 - 1) * (p_2 - 1) * \dots\dots\dots\dots * (p_r - 1)$

$\phi(N) = \prod_{i=1}^{r}(Pi - 1)$

$choose\ an\ integer\ e,\ 1 < e < \phi(N)\ such\ that$

$gcd(e, \phi(N)) = 1$

$the\ pair\ (N, e)\ is\ the\ public\ key$

$compute\ the\ unique\ d\ such\ that$

$ed = 1\ mod\ \phi(N)$

i.e. $d = e^{-1}\ mod\ \phi(N)$

$private\ key\ is\ the\ pair\ (N, d)$

Step 2: Digital Signature

$Use\ MD5\ algorithm\ to\ generate\ message\ digest\ of\ the\ document\ to\ be\ send$

$Let\ an\ integer\ M\ reperesent\ the\ digest\ generated$

$Generate\ S\ (which\ is\ the\ digital\ signature)$

$S = M^d\ mod\ N$

Step 3: Encryption

$For\ any\ message\ m$

$The\ cipher\ c, is\ computed\ using\ the\ public\ key\ (N, e)$

$c = m^e\ mod\ N$

Step 4: Decryption

$For\ any\ cipher\ text\ c$

$The\ plain\ text\ is\ recovered\ using\ private\ key\ (N, d)$

$Compute\ m$

$m = c^d \bmod N$

Step 5: Signature Verification

$To\ verify\ the\ signature, an\ integer\ V\ is\ generated\ using\ public\ key\ (N, e)\ and\ the\ digital\ signature\ S$

$V = S^e \bmod N$

$Use\ MD5\ algorithm\ to\ extract\ the\ message\ digest\ M1\ from\ integer\ V.$

$Then\ compute\ message\ digest\ M2\ from\ the\ signature\ S.$

$Test\ if\ M1\ and\ M2\ are\ equal\ (M1 == M2?)\ then\ the\ signature\ is\ valid$

$Else, invalid.$

Performance Analysis of Multi- Level Cryptographic Algorithm

The algorithm was implemented in Java programming language and it was evaluated using two parameters; Execution time and Memory space.

Execution Time

The execution time considered as the time that the algorithm takes to produces a cipher text from a plain text. It is commonly calculated by counting the total operations performed by the system where each operation takes a fixed amount of time. An algorithm performance time may vary with different input size (Khatoon and Ikram, 2014). Execution time normally increases as the input size increase. A good algorithm is expected to execute in a very small period of time

Memory Space

Apart from Execution time, Memory space is also an important measure to judge the performance of an algorithm. It is the amount of memory which the algorithm needs for performing its computations. A good algorithm keeps the amount of memory as small as possible. The way in which the amount of storage space required by an algorithm varies with the size of the problem it is solving (Khatoon and Ikram, 2014). However, the following table shows the execution time and the memory space required by the algorithm

Table 1: Execution Time and Memory Space of the Hybrid Algorithm.

Input Size	Time (millisecond)	Memory Space (Kilobyte)
7	0.7	0.39
31	0.9	1.36
169	1.6	5.70
869	2.1	23.90
1238	2.4	40.10

Comparison of Performance between the Multi-Level Cryptographic Algorithm and Existing Algorithms

To find out the efficiency of this algorithm, the algorithm was evaluated using two parameters; execution time and memory space, the algorithm was then compared with some existing algorithms based on this parameters and obtained the following:

Execution Time (millisecond)

Table 2: Comparison of Execution Time.

Input Size (No. of Characters)	Execution Time (Millisecond)		
	Multi-level Algorithm	Multi- prime Algorithm	MD5 Algorithm
7	0.7	0.9	0.9
31	0.9	1.3	1.7
169	1.6	2.1	3.6
869	2.1	3.8	4.9
1238	2.4	4.7	6.2

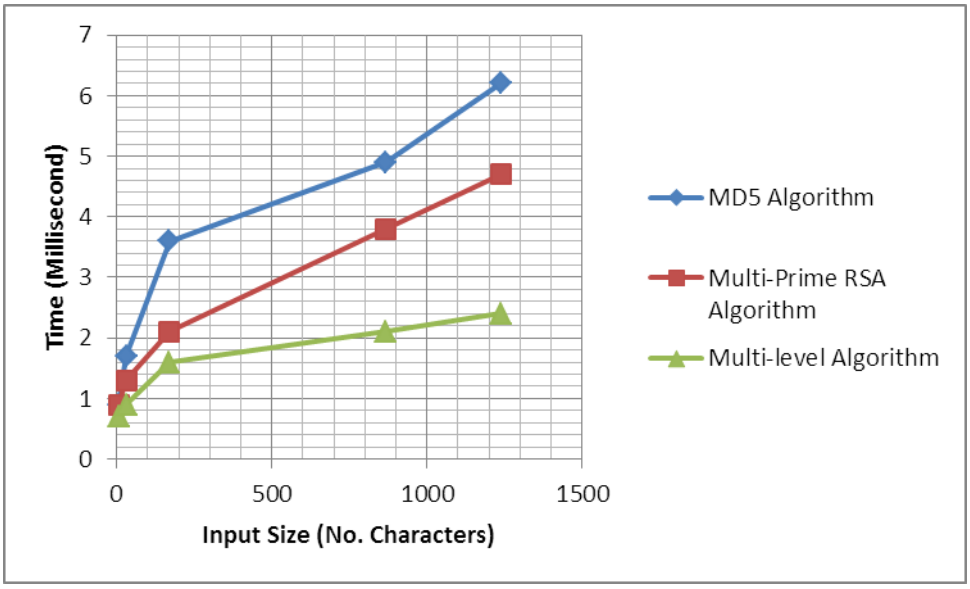

Fig. 4: Graph of Comparison Based on Execution Time.

Memory Space (KB).

Table 3: Comparison of Memory Space.

Input Size (No. of Characters)	Memory Space (Kilobyte)		
	Multi-level Algorithm	Multi- prime Algorithm	MD5 Algorithm
7	0.39	1.86	1.24
31	1.36	1.86	2.76
169	5.70	7.90	8.98
869	23.90	43.21	25.76
1238	40.10	81.87	84.32

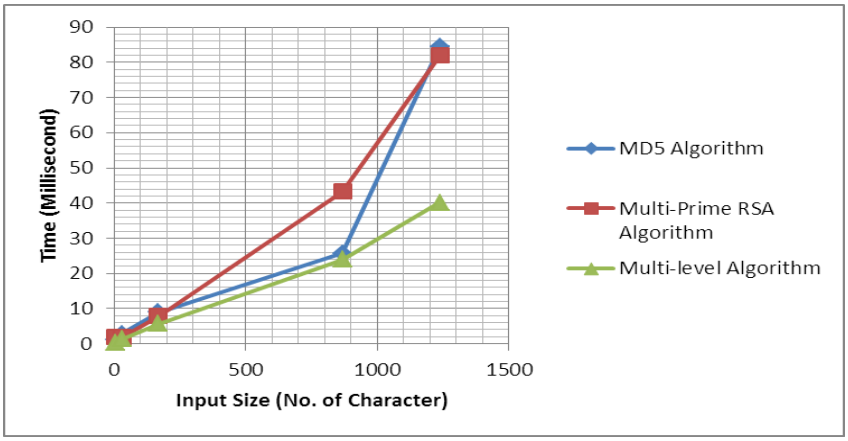

Fig. 5: Graph of Comparison Based on Memory Space.

CONCLUSION

Multi-level cryptographic algorithm was developed in order to overcome the shortcomings of the previously used cryptographic algorithms (such as MD5 and Multi-prime RSA), the algorithm was designed, evaluated and compared its performance with the existing algorithms. It was found out that, the algorithm is very fast in execution time and required less memory space compared to other algorithms. Therefore the algorithm is very efficient and can resist some kinds of attacks like brute force and mathematical attacks.

REFERENCES

1. Ahmed, I. and Husain, R Design of Muli-level Algorithm for Data Storage Security in Cloud Computing, WJERT, 2017; 3(4): 112-119.
2. Caesar G. J and Kennedy, J. F Cryptography. Security Engineering: A Guide to Building Dependable Distributed Systems, 2017.
3. Khatoon, A. and Ikram, A.A Performance Evaluation of RSA Algorithm in Cloud Computing Security. International Journal of Innovation and Scientific Research, 2014; 12(1).

4. Padmavathi, B. and Kumari, S. R A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique. International Journal of Science and Research. India [Online] ISSN, 2013; 2319-7064.
5. Roy, R. C Security in Cloud Computing. International Journal of Computer Applications, 2014; 96(15).
6. Sudhansu, R. J. and Nayak, B. Enhancing data security in cloud computing using RSA encryption and MD5 algorithm. International Journal of Computer Science Trends and Technology, 2014; 2(3).
7. Sugunya, M., Boopal, M. E and Naveena, M. Implementing multiprime RSA algorithm to enhance the data security in cloud computing. International Journal of Innovative Research in Scienec, Engineering and Technology, 2015; 4(3).
8. Tripathi, R and Agrawwal, S Comparative Study of Symmetric and Asymmetric Cryptography Techniques. International Journal of Advance Foundation and Research in Computer, 2014; 1(6).
9. Zareen Enhancement on implementation of multi-prime and multi-power RSA algorithm. Department of computer Science and Engineering, Thapar University, 2011.