International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12 – No. 4, July 2017 – www.ijais.org

Effective Cryptographic Technique for Securing Cloud Storage Systems

Isaac Kofi Nti
Department of Electrical & Electronic Engineering
Sunyani Technical University
Sunyani, Ghana

Eric Gyamfi
Department of Electrical & Electronic Engineering
Sunyani Technical University
Sunyani, Ghana

Marvin Appiah Osei
Department of Electrical & Electronic Engineering
Sunyani Technical University
Sunyani, Ghana

ABSTRACT
Storing of data in the cloud (Cloud Computing) offers an effective and quick way of granting access to ones information from a third party service provider, providing business expansion at a lesser cost. Cloud data storage systems provide means to store bigger data in storage servers [1]. The data stored in the cloud are stored and accessed from the internet over a longer time, making data exposed to hackers to steal stored and transmitted data over the cloud environment, leading to data integrity loss and users of cloud data unhappy. This paper proposed a novel cryptographic techniques to enhance the security in cloud environment and reduce the time associated with cryptographic encryption to a minimum.

General Terms
Cryptographic Technique, Securing Cloud Storage Systems

Keywords
Cloud-Storage, Cryptographic-Tactics, Data Confidentiality, Data Integrity, Data Storage

1. INTRODUCTION
The need for storage system for companies, businesses and enterprise grows 50% approximately in every year, this leads to a huge investment in storage facilities by organizations which are underuse. On the other hand there is a high cost associated with huge data management. Meanwhile every business man wants to minimize risk and maximise profit, hence to overcome the huge cost associated with data management a lot of small and medium size organizations ends up in outsourcing the storage of their organization data to 3rd party storage service providers that offer storage management services and on-demand storing space [2].

Computing in the cloud (cloud computing) is among the latest method in current dispensation for minimizing operation cost and losses in today's business world and information communication and technology (ICT) age. Storage in the cloud provides and convenience means of data shearing to cloud users, cloud users can remotely store and retrieved data in the cloud at easy [1]. Protecting data integrity in the cloud has been a concern, because the physical possession of the outsourced data is not known to the user. Computing in the cloud permits cloud users to store their data in space so as to make use of accessible on-demand services. Cloud computing allows and offers small and medium scale businesses with limited resources and budgets to achieve high savings and improvement in productivity by employing cloud based services such as project management for enhancing collaboration among staff members.

One of the key concern in cloud computing adaptation is security. Cloud data storage increases every day, hence a secured mechanism is required to that data stored in the cloud is secured from unauthorised access [1]. Security is key in data stored in cloud environment [2] [3, 1]. The advancement of outsourced storage to storage service dealers point out the importance of mounting economical and efficient security algorithms and methods to safeguard the information hold on in an exceedingly networked storage system [2].

Ensuring the integrity of data distributed over cloud is a very difficult challenge and the key solution to this challenge is to employ cryptographic methods in cloud environs [1]. In this paper we look at various cryptographic techniques proposed by other researchers and propose an effective cryptographic technique for securing data stored in the cloud, which will bring improvement in current implementations of cryptographic file systems: data veracity, key management for cryptographic file systems by implementing lazy revocation, and constancy of encrypted file objects.

2. CRYPTOGRAPHIC CLOUD STORAGE
Cryptography (or cryptology) stand for "hidden secret" is the practice and study of techniques for secure communication within the presence of third parties (called adversaries). A lot of usually, it is concerning constructing and analysing protocols that overcome the influence of adversaries and that are associated with many aspects in data security like information confidentiality, information integrity, authentication, and non-repudiation [4]. The information stored might be make known or altered by any unauthorized access, it is therefore vital to ensure that the user's sensitive data are secured. Data storage in cloud must be secured [5], hence cryptographic techniques is adopted for data security in the cloud. Figure 1 shows the cloud storage strategy.

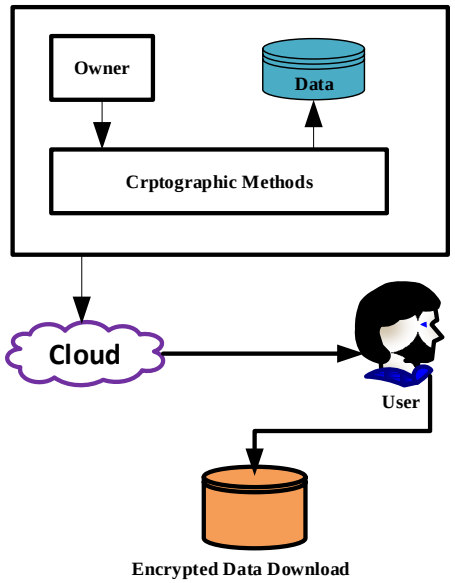


Figure 1 Cloud Environment Strategy

Figure 1 shows cryptographic cloud storage, the data owner secures the sensitive information from unlawful access (hackers) by applying cryptographic methods to the sensitive data. The encrypted data is then uploaded to the cloud environs. The data is then decrypted and downloaded by the authorized user. Two factors namely Integrity and Confidentiality measure the strength of Cryptographic Cloud Storage [1]. Confidentiality in cloud storage Cryptographic means the user information or data is encrypted with the advanced cryptographic techniques, which maintain the secrecy of data [6]. Integrity in cloud storage is assurance that the stored information in the cloud is and cannot be modified by unauthorized people

2.1 Cryptographic Techniques

The main parts of a cryptographic storage service which might be enforced by employing a completely different techniques, out of that, some were aimed specially for cloud storage. Within the starting of the Computing in the Cloud, common secret writing technique similar to Public Key secret writing was applied. This ancient methods doesn't offer the anticipated result because it support one to at least one secret writing sort of communication. Public Key secret writing isn't extremely climbable. This gave rise to manoeuvre onward to more advanced secret writing strategies. The advanced cryptographic strategies comprises the below secret writing strategies.

- ✓ Searchable Encryption
  - ➢ Asymmetric Searchable Encryption (ASE)
  - ➢ Symmetric searchable encryption
- ✓ Identity Based Encryption
- ✓ Homomorphic Encryption
- ✓ Cloud DES Algorithm
- ✓ Attribute-based Encryption
  - ➢ KP-ABE
  - ➢ MA-ABE
  - ➢ CP-ABE

2.1.1 Searchable Encryption

Searchable symmetric encryption (SSE) allows an organization to source the storage of their data to a unique party in an exceedingly very personal manner, whereas maintaining the pliability to by selection search over it. This disadvantage has been the main focus of active analysis and lots of security definitions and constructions are planned. Private-key storage outsourcing [7] permits shoppers with either restricted resources or restricted experience to store and distribute large amounts of symmetrically encrypted info at low worth. Since regular private-key secret writing prevents one from searching over encrypted info, purchasers jointly lose the facility to selectively retrieve segments of their info. To influence this, several techniques area unit projected for provisioning bilaterally symmetric cryptography with search capabilities [7]; the ensuing construct is typically referred to as searchable cryptography. The house of searchable cryptography has been known by office together of the technical advances, which is able to be accustomed balance the requirement for every privacy and national security in information aggregation systems.

2.1.2 Symmetric Searchable Encryption (SSE)

It is acceptable for the environs wherever a client that searches the information is accountable for the information generation. One Writer/Single Reader (SWSR) comes from cloud storage word. SSE techniques were given in [8] and magnified constructions and security terms [7]. SSE has 2 major edges they're efficiency and security, even though it has disadvantages like usefulness and trade-off efficiency, SSE techniques are acceptable for the individual that perform the encryption and for the person that searches with a keyword from the cloud storage system. SSE are mostly economical because, pseudo-random functions and as well as block ciphers for encryption purpose. In [7], the researchers affirms that search technique is efficient and effective, since SSE permits pre-processed data through efficient represent in information structures.

2.1.3 Homomorphic Encryption

The Homomorphic encryption concepts explain by Ronald Rivest et al. cited by [9]. This encryption technique is useful in the cloud environs to protect the records. The Homomorphic encryption Homomorphic encryption concepts permits performing computations on the encrypted data. It is only of the advanced cryptographic technique. The key drawback of homomorphic encryption is given [10] as been slow with respect to processing time during computation.

2.2 RELATED WORK

2.2.1 Cryptographic file systems Integrity

Most existing cryptographic file systems shows that there is a quid pro quo in the middle of the amount of server-side storing of integrity data and the access period to read and write discrete file chunks [2]. ECFS proposed by [11], TCFS proposed by [12], these two are an advance of [13], NASD proposed by [14] and SNAD are the most universally integrity cryptographic techniques. This approaches stores a keyed hash or a hash for every blocks on the server, their output is a linear storage for integrity in the numeral of chunks at the storage server and unchanged access period. Cepheus by [15] and SUNDR by [16] proposed a methodology where for every file, a Merkle hash tree plotted on the ith-node tree of the file. The authentication root of the hash key is only stored on the storage server, which contributes to an increase in the period involved in checking the integrity of chunks (blocks) and

chunks content update. Another approach called SIRIUS proposed by [17] provides a storage of digital signature for every file, making the entire file to be read whiles checking the integrity of individual block in the file.

2.2.2 Authenticated Encryption

An authenticated encryption proposed by [18, 19] makes use of a cryptographic basic that offers communication and privacy authenticity at the same period. The customary tactic for building true encryption is by all-purpose configuration, thus, a mishmash of an unforgeable communication (message) authentication code (MAC) and a secure encryption pattern. On the other hand, [18] did an analysis of the composition security and made available proofs, that some of the generally whispered secure compositions are in fact insecure. It believed the authenticated encryption method employed in SSL and SSH is evidenced to be insecure by Krawczyk (2001) cited by [2] and [20], respectively. An advance integrity mechanism was proposed by [20] that offer protection against repeat and out-of-order transfer attacks for network protocols. We make focus on the storage scenario integrity which makes a difference from the network case proposed by [20].

A parallel computing as a means to enhance the performance of cryptography was proposed by [21], his proposed method addresses the matter of enhancing the performance of robust cryptographic algorithms that are normally used and executed by most internet users.

Another methodology to reduce the encryption speed was proposed by [22]. In their system, a combinations the benefits of multiprocessing and cryptological algorithms was used. The employment of multiprocessing enhances the speed of system when compared to the normal crypto systems. During this approach they have divided a file into two slices and have applied one rule with totally different key for every slice and the processing of the algorithm is done in a parallel environment. From the experiments it's distinguished that the execution time of a cryptological rule is significantly reduced during a parallel environment in comparison to the generic ordered ways.

Hur, explains the cryptographically based mostly solution for data sharing mistreatment cipher-text policy attribute-based encryption (CP-ABF) to boost the security of the data. In this technique the data owners defines the access policies on the information to be distributed [23]. The foremost downside of this technique is that the unauthorized users will access the key to decipher the encrypted information.

The above discussion points outs that there is a need for a faster and efficient crypto algorithm for present applications. This paper therefore propose a cryptography algorithm that breaks every file (message) into three separate chunks and perform parallel encryption to reduce the encryption time.

3. METHODOLOGY

The encryption is a process of making plaintext to cipher-text and decryption is converting cipher-text to plain text. Performing these processes in the same in sequential way is always consumes a lot of time. The proposed method thus the implementations of the encryption and decryption in parallel way using threads. A subdivision approach is used to divide the plaintext into N number of chunks with equal size, each chunk is then moved to the threads, which each threads takes these chunks of the data as input and then encrypts them and present the encrypted text as outputs. All the various outputs are put together to form a single file as shown in figure 2.

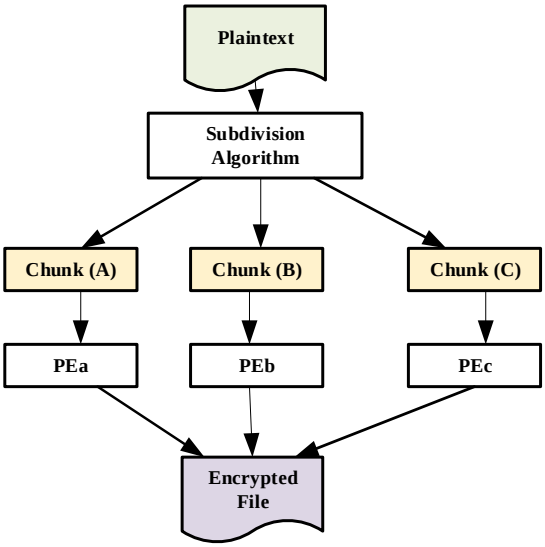


Figure 2 Parallel Encryption Phase

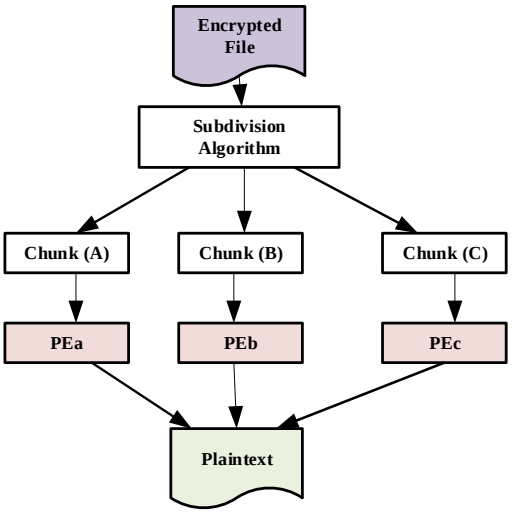


Figure 3 Parallel Decryption phase

The approach is tested on Caesar cipher method and transposition method on text file. The implementation is done on sequential and parallel way for the same input file. The key selection is done using user input. For encryption process equation 1 is use, whiles decryption process equation 2 is used. Both equation 1 and 2 shows how the encryption and decryption are done using single key on the given input text data.

$$\text{C p} \quad = \quad \text{Plain text} + \text{K c} \quad (\text{Key}) - - - - - (1)$$

$$\text{Plaintext} = \text{C p} + \text{K c} \quad (\text{Key}) - - - - - (2)$$

The implementation steps for encryption is as given bellow.

- ✓ Read the given plaintext file.
- ✓ Split the file into number of chunks.
- ✓ Create the threads and assign each chunks to the created threads.
- ✓ All the chunks are encrypted in parallel using threads.

- ✓ Write the result to a new file which is an encrypted file.

The parallel implementation steps for decryption are as given bellow.

- ✓ Read the given cipher text file.
- ✓ Split the file into number of chunks.
- ✓ Create the threads and assign each chunks to the created threads.
- ✓ All the chunks are decrypted in parallel using threads.
- ✓ Write the result to a new file which is a decrypted file.

4. RESULTS AND DISCUSSIONS

The sequential process of encryption take time [21] as the size of file increases. In this work same sequential process of encryption was implemented by means of threads base on parallel for loop, and it was observed that, the performance (speed) improved compared to the traditional sequential method. The results proves that using threads execution reduces time to a half required in traditional sequential execution of the same content (plaintext) seize, but as the number of threads increases, the performance will be degraded.

4.1 Encryption phase

Figure 4 shows a sample plaintext to be fed into the proposed algorithm subdivision section and figure 5 shows the output of the subdivision, which the input plaintext is subdivided into three equal chunks, which are supplied to parallel encryption process.

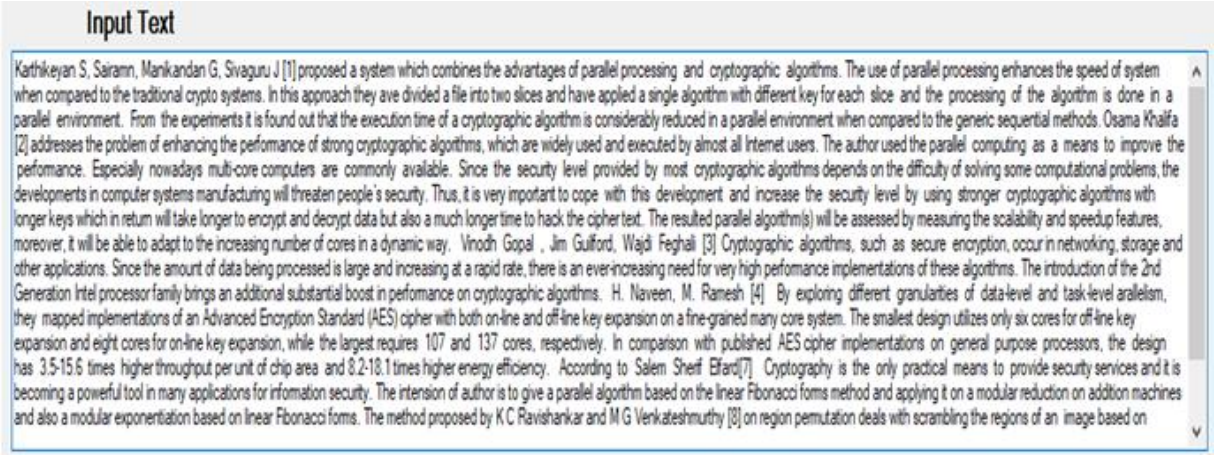


Figure 4 Sample Plaintext

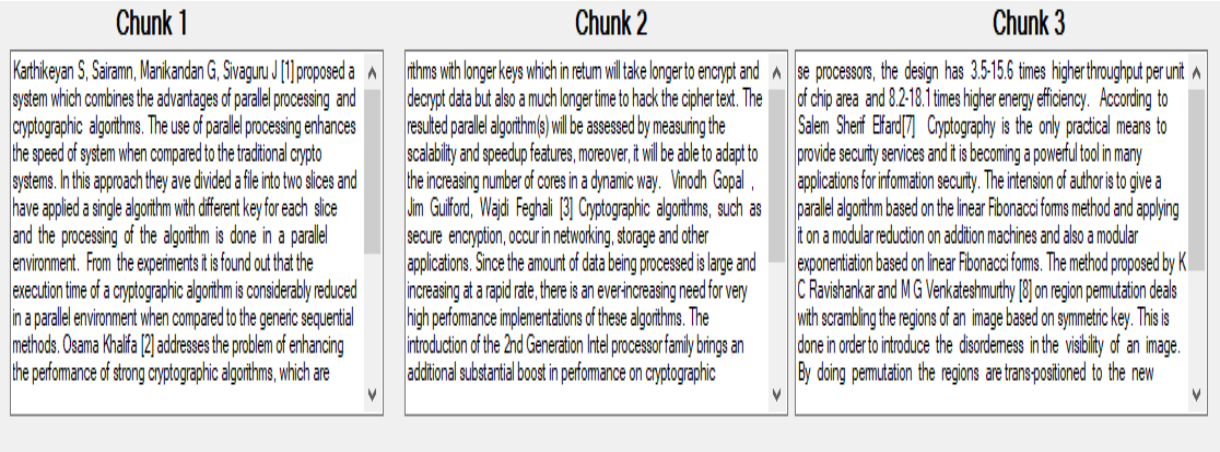


Figure 5 Input File Break into Chunks File

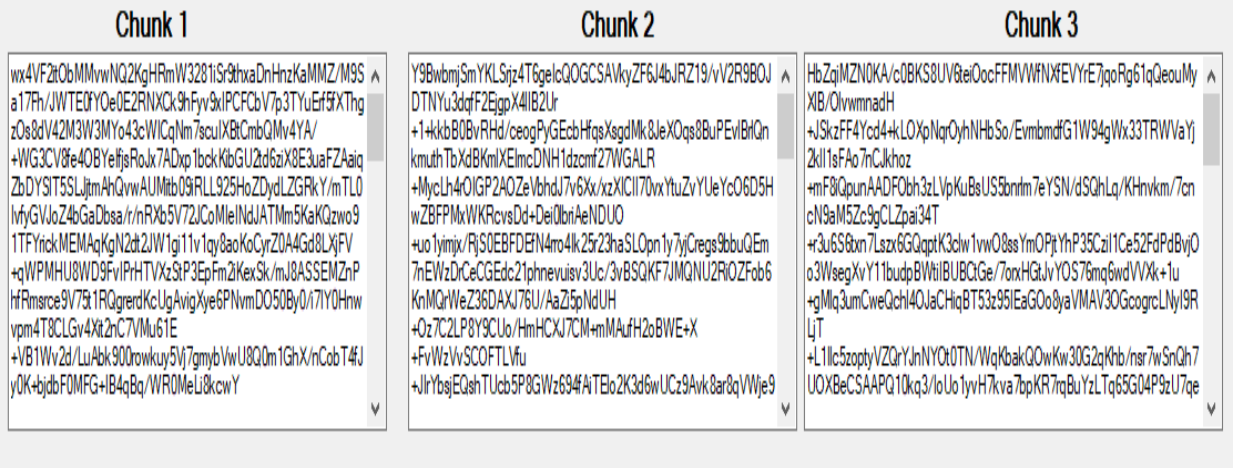


Figure 6 The Encrypted Output of each chunk

Figure 7 Combined Encrypted File

4.2 Decryption Phase

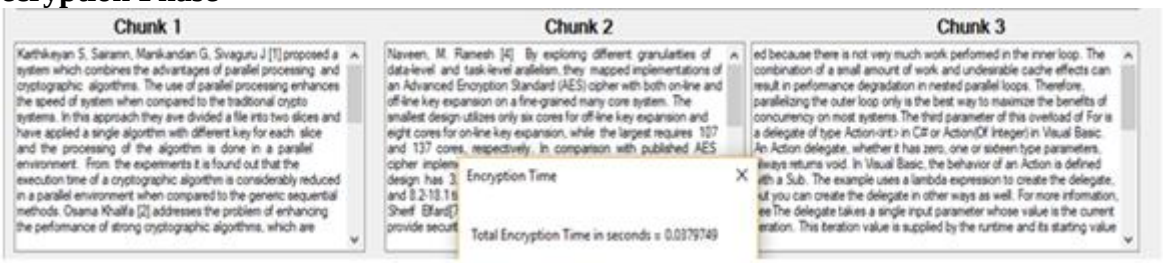
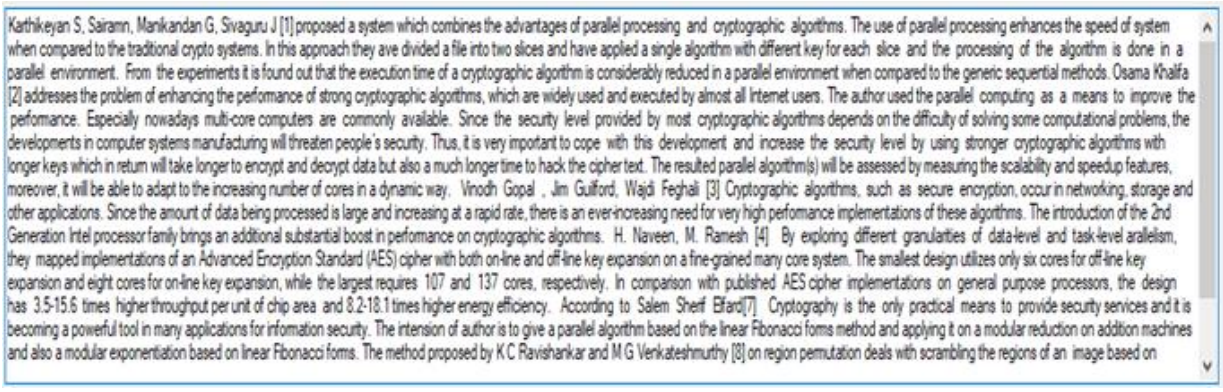


Figure 8 Decrypted Output of each Chunk File



Karthikeyan S, Sairamn, Manikandan G, Sivaguru J [1] proposed a system which combines the advantages of parallel processing and cryptographic algorithms. The use of parallel processing enhances the speed of system when compared to the traditional crypto systems. In this approach they ave divided a file into two slices and have applied a single algorithm with different key for each slice and the processing of the algorithm is done in a parallel environment. From the experiments it is found out that the execution time of a cryptographic algorithm is considerably reduced in a parallel environment when compared to the generic sequential methods. Osama Khalifa [2] addresses the problem of enhancing the performance of strong cryptographic algorithms, which are widely used and executed by almost all Internet users. The author used the parallel computing as a means to improve the performance. Especially nowadays multi-core computers are commonly available. Since the security level provided by most cryptographic algorithms depends on the difficulty of solving some computational problems, the developments in computer systems manufacturing will threaten people's security. Thus, it is very important to cope with this development and increase the security level by using stronger cryptographic algorithms with longer keys which in return will take longer to encrypt and decrypt data but also a much longer time to hack the cipher text. The resulted parallel algorithm(s) will be assessed by measuring the scalability and speedup features, moreover, it will be able to adapt to the increasing number of cores in a dynamic way. Vinodh Gopal , Jim Guilford, Wajdi Feghali [3] Cryptographic algorithms, such as secure encryption, occur in networking, storage and other applications. Since the amount of data being processed is large and increasing at a rapid rate, there is an ever-increasing need for very high performance implementations of these algorithms. The introduction of the 2nd Generation Intel processor family brings an additional substantial boost in performance on cryptographic algorithms. H. Naveen, M. Ramesh [4] By exploring different granularities of data-level and task-level arallelism, they mapped implementations of an Advanced Encryption Standard (AES) cipher with both on-line and off-line key expansion on a fine-grained many core system. The smallest design utilizes only six cores for off-line key expansion and eight cores for on-line key expansion, while the largest requires 107 and 137 cores, respectively. In comparison with published AES cipher implementations on general purpose processors, the design has 3.5-15.6 times higher throughput per unit of chip area and 8.2-18.1 times higher energy efficiency. According to Salem Sherif Elfard[7] Cryptography is the only practical means to provide security services and it is becoming a powerful tool in many applications for information security. The intension of author is to give a parallel algorithm based on the linear Fibonacci forms method and applying it on a modular reduction on addition machines and also a modular exponentiation based on linear Fibonacci forms. The method proposed by K.C Ravishankar and M.G Venkateshmurthy [8] on region permutation deals with scrambling the regions of an image based on

Figure 9 Combine Output of Decrypted File

Figure 6 shows the output of each encrypted chunk file, the various outputs are combined to give a completed encrypted file of the input plaintext as shown in figure 7.

Figure 8 and 9 shows the output files obtained after decryption, the complete encrypted file is broken into chunks and each chunk file decrypted separately and the results combined to form one file as shown in figure 9.

4.3 Discussions

The execution time is used as the basic for ascertaining the performance of AES parallel and natural for loop algorithm [24]. Thus the performance of an encryption algorithm is inversely proportional to the time taken to encrypt the file. The lesser the time required for execution, the higher the performance of the encryption algorithm. Table 1 and 2, figure 9, 10, 11 and 12 shows the time involved in the encryption and decryption methods of the proposed framework and the traditional sequential algorithm.

From table 1 it is seen that using the proposed method to encrypt a data of 6Kb is 0.0484068 seconds faster than the traditional sequential execution algorithm, a 0.0307668 seconds faster when input plaintext is 7Kb and 0.0420913 seconds faster for 9Kb input plaintext.

Table 1: Encryption with time performance

Serial No.	Data Size	Encryption Time with parallel Algorithms	Encryption Time with natural for loop Algorithms	Variance
1	6 KB	0.0365364	0.0849432	0.0484068
2	7 KB	0.0342448	0.0650116	0.0307668
3	9 KB	0.0381994	0.0802907	0.0420913

Table 2: Decryption of Cypher text with time

Serial No.	Data Size	Decryption Time with parallel Algorithms	Decryption Time with natural for loop Algorithms	Variance
1	8 KB	0.0293788	0.0681649	0.0387861
2	9 KB	0.0281139	0.0644594	0.0363455
3	11 KB	0.0347711	0.0625541	0.027783

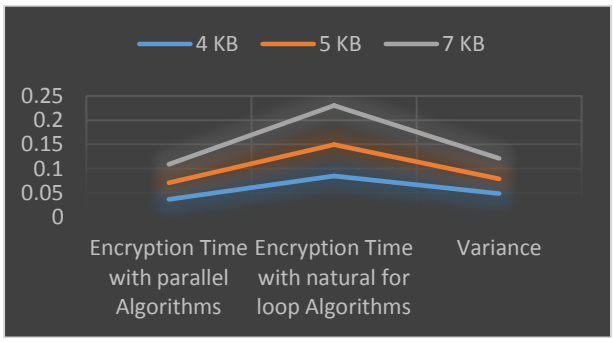


Figure 9 Encryption Time Between parallel and for loop Algorithms

Figure 10, shows the performance for the encryption and decryption algorithms with different data size.

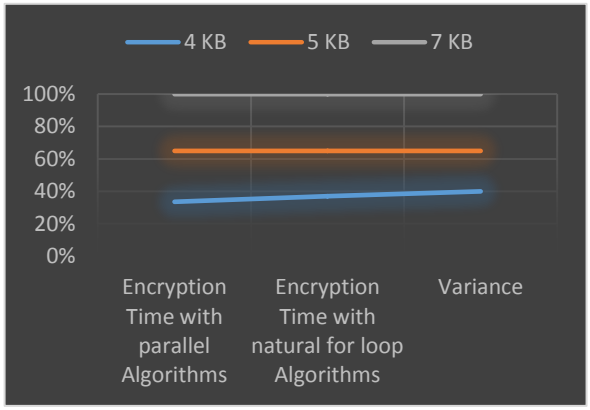


Figure 10 Performance Analysis of encryption time

In figure 11, the various execution time for decryption of the cypher text is shown.

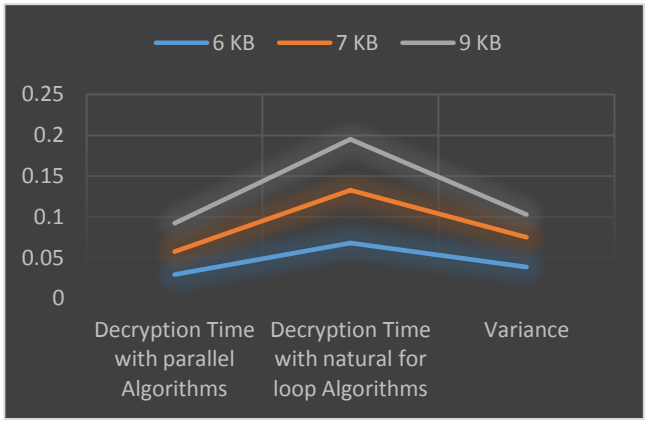


Figure 11 Decryption execution time of cypher text

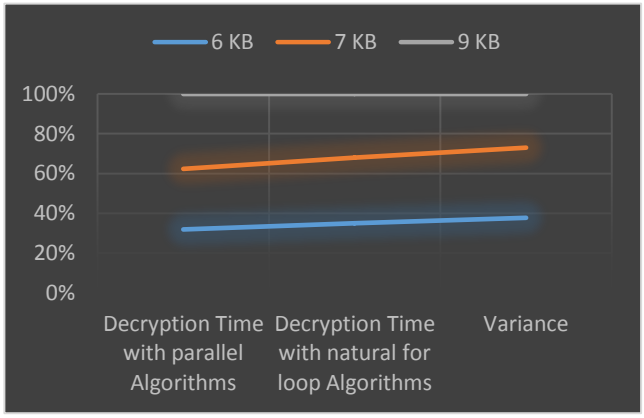


Figure 12 Performance Representation of Execution Time

5. CONCLUSION

The paper presents a parallel for loop algorithm and a comparative analysis with a natural for loop algorithm for data encryption and storage in the cloud based on different set of parameters. The results obtain reveals that parallel encryption algorithm is highly efficient algorithm with a high value throughput and performance based on execution time, and it is highly secured (data integrity and confidentiality) and the best power effective algorithm as compared natural for loop. It is a high speed algorithm nonetheless it is cryptographically secure.

6. ACKNOWLEDGMENTS
Our thanks to the almighty God for protection and guidance and to all who have contributed towards development of the research.

7. FUTURE WORKS
This paper offers application of encryption and decryption algorithm for text file employing diverse cryptographic methods using C# as programming language. The encryption and decryption are implemented for Caesar cipher and subdivision algorithm. The time involved with sequential and parallel methods proposes that, employing threads, it is possible to realize parallelism to enhance the performance of encryption algorithms. The same comparison may be done for different algorithms and for different input formats in feature.

8. REFERENCES
[1] R. Kirubakaramoorthi, D. Arivazhagan and D. Helen, "Survey on Encryption Techniques used to Secure Cloud Storage System," correspondenceIndian Journal of Science and Technology, vol. 8, no. 36, pp. 1-7, 2015.

[2] A. M. Oprea, Efficient Cryptographic Techniques for Securing Storage Systems, School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, 2007.

[3] D. Patil, R. Bhavsar and A. Thorve, "Data security over cloud," in Emerging Trends in Computer Science and Information Technology (ETCSIT2012) , 2012.

[4] M. Tahghighi, S. Turaev, R. Mahmod, A. Jafaar and M. Said, "The Cryptanalysis and Extension of the Generalized Golden Cryptography," in IEEE conference on open system, Lankawi, Malaysia,, 2011.

[5] A. Bessani, M. Correia and B. Quaresma, "DEPSKY: dependable and secure storage in a cloud-of-clouds.," in 6th Conference on Computer Systems (EuroSys'11), 2011.

[6] S. Yu, C. Wan, K. Ren and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," in IEEE Communications Society for publication, 2010.

[7] C. Reza, G. Juan, K. Seny and O. Rafail, "Searchable Symmetric Encryption: Improved defination and efficient construction," 2006. [Online]. Available: https://eprint.iacr.org/2006/210.pdf. [Accessed 12 March 2016].

[8] D. Wagner, D. Song and A. Perrig, "Practical techniques for searching on encrypted data," in IEEE Symposium on Research in Security and Privacy, 2000.

[9] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," in Proceedings of Cryptography LNCS, 2001.

[10] C. Fontaine and F. Galand, "A survey of homomorphic encryp¬tion for nonspecialists," EURASIP Journal on Information Security, pp. 1-15, 2007.

[11] D. Bindel, M. Chew and C. Wells, "Extended cryptographic file system," manuscript, 1999.

[12] G. Cattaneo, L. Catuogno and A. P. Sorbo, "The design and implementation of a transparent cryptographic file system for Unix," in USENIX Annual Technical Conference 2001, 2001.

[13] M. Blaze, "A cryptographic file system for Unix," in First ACM Conference on Computer and Communication Security (CCS), 1993.

[14] H. Gobioff, D. Nagle and G. Gibson, "Integrity and performance in network-attached storage," 1998.

[15] K. Fu, Group sharing and random access in cryptographic storage file systems, Master's thesis, Massachusetts Institute of Technology (MIT), 1999.

[16] J. Li, M. Krohn, D. Mazieres and D. Shasha, "Secure untrusted data repository," in 6th Symposium on Operating System Design and Implementation (OSDI), 2004.

[17] E. Goh, H. Shacham, N. Modadugu and D. Boneh, "SiRiUS: Securing remote untrusted storage," in Network and Distributed Systems Security (NDSS) Symposium, 2003.

[18] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," in Asiacrypt 2000, 2000.

[19] J. Katz and M. Yung, "Unforgeable encryption and chosen ciphertext secure modes of operation," in FSE 2000, 2001.

[20] M. Bellare, T. Kohno and C. Namprempre, "Authenticated encryption in SSH: Provably fixing the SSH binary packet protocol," in 9th ACM Conference on Computer and Communication Security (CCS), 2002.

[21] O. Khalifa, "The performance of cryptographic algorithms in the age of Parallel computing," Heriot Watt University School of Mathematical and Computer Science, 2011.

[22] S. Karthikeyan, Sairamn, G. Manikandan and J. Sivaguru, "A Parallel Approach for Improving Data Security," Journal of Theoretical and Applied Information Technology, vol. 39, no. 15, pp. 1-7, 2012.

[23] J. Hur, " Improving Security and Efficiency in Attribute-Based Data Sharing," In IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 10, pp. 2271-2282, 2013.

[24] O. Nyarko- Boateng, M. Asante and I. K. Nti, "Implementation of Advanced Encryption Standard Algorithm with Key Length of 256 Bits for Preventing Data Loss in an Organization," International Journal of Science and Engineering Applications, vol. 6, no. 03, pp. 88-94, 2017.