

Data Security in the Cloud Using Serpent Encryption and Distributed Steganography

Izevbizua, Peter Odion

Department of Computer Science,
University of Port Harcourt,
Nigeria.

Email: peter.izevbizua@uniport.edu.ng

Abstract

Despite the enormous benefits derived from the adoption of cloud computing concept, its widespread acceptance has been considerably encumbered by security concerns. The enlarged attack surface in a cloud environment makes it more vulnerable to existing and emerging security threats. Conventional data security approaches have been found incapable in curtailing these threats and this unpleasant trend has necessitated the need for a futuristic approach to data security. Serpent encryption algorithm and distributed steganography are already proven techniques for securing data. This paper proposes an enhanced mechanism to ensuring data security by strategically combining serpent cryptographic algorithm and distributed steganography. This unified approach leverages on the strength of these two proven techniques to achieve a robust mechanism for ensuring confidentiality and integrity of data in the cloud.

Keywords: Cloud Computing; Cryptography; Steganography; Data Security.

1.0 Introduction.

Cloud computing is a revolutionary internet-enabled computing technology that has progressively developed in recent years. Mell & Grance (2011) aptly defined cloud computing as “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. Literally, the idea behind cloud computing is to enable clients (e.g., users, or institutions) to access computing resources via the internet and pay per use as utilities, the same way users normally pay for water, electricity and related utility services (Alshuwaier et al., 2012). In as much as cloud computing concept affords enormous benefits to both consumers and the service providers, its adoption/acceptance has been significantly encumbered primarily by security concerns and some other challenges which include service quality, performance and integration. The need to ensure security of data in the cloud is imperative and this has motivated researchers within the computing community to invent mechanisms to ensure the confidentiality, integrity and availability of data in the cloud (Garini et al. 2014).

From the standpoint of data security which has been an important aspect of quality of services, cloud computing unavoidably poses new challenging security threats for a number of reasons which are centered on the peculiarity of cloud computing concept. Cloud data security aims to ensure data is adequately secured for the entire duration of its life-cycle (data-at-rest; data in transit within communication channels; as well as data in use). Consequently, combating these

threats would require more than the traditional cryptographic primitives (Mrinal and Trijit, 2014). In the light of the foregoing, this paper proposes a unified approach to ensuring security of data in the cloud using a combination of the serpent cryptographic encryption scheme and distributed steganography.

2.0 Review of Cloud Computing Concept and Data Security.

This section attempts to briefly review cloud computing concept, its associated security risks, and existing mechanisms for data security.

2.0.1 Cloud Computing Model.

Cloud computing is basically an internet-based technology for providing configurable computing resources (such as networks, servers, and storage services) using flexible infrastructure. It is a model of network computing in which virtualization technology is leveraged to make programs or applications run on one or more connected servers rather than a local computing device as is the case with traditional client-server computing model.

The National Institute of Standards and Technology (NIST), an international research group led by scientists at the U.S Department of Commerce categorized Cloud computing into three service model and four deployment model. NIST also offered five essential attributes/characteristics of Cloud computing (Taniya, 2013). The service model is comprised of Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The basement of this logical stack of functionality is Infrastructure as a Service which only offers hardware services to consumers. Here, the consumers are only meant to utilize infrastructure like storage, servers and networking devices. In the middle layer lies Platform as a Service which offers consumers an application development environment to enable consumers create their own applications. Software as a Service resides atop the stack. In this layer, the consumers run online software applications provided by service vendors and on a pay-as-you-go basis and therefore do not have to engage in software installation and setup in their own computing devices.

Taniya (2013) maintained that NIST's cloud deployment models include: (1) Private cloud: refers to a cloud architecture built for the exclusive use of one client. It's architecture provides hosted services for exclusive use by a single organization comprising multiple consumers protected by a firewall. (2) Public cloud: These are designed for open use by the general public with access control mechanisms provided by third-party vendors. They are usually hosted on the premises of the cloud provider. (3) Hybrid cloud: This is a composition (or inter-operation) of two or more distinct cloud infrastructures (private, community, or public. (4) Community cloud: This is controlled and used by a group of organizations that have shared interests.



The five essential attributes of cloud computing as propounded by NIST are: Rapid Elasticity, Measured Service, On-Demand Self-Service, Ubiquitous Network Access and Location-Independent Resource Pooling/Multi-tenancy.

2.0.2 Cloud Data Security.

The fact that virtualization is the underlying technology leveraged in the operation of cloud environment poses increased vulnerabilities to the cloud. Messer (2012) stated that virtualization is the creation of virtual resources from physical resources. In a virtual environment, one host that previously ran a single operating system now has the ability to run multiple guest operating systems. A major drawback of virtualization as well as the cloud environment is that the increased attack surface poses increased opportunities for inherent vulnerabilities to be exploited.

2.0.3 Responsibility for Cloud Security.

The enlarged attack surface inherent in cloud surface makes it difficult to determine who is particularly responsible for securing various software and hardware components that constitute the cloud service and deployment models. Robert (2013) maintained that cloud service providers (CSP) tend to place responsibility for securing the cloud data on the clients but many clients assume security responsibilities are entirely provided by the CSPs. He stressed the need to clarify the shared security responsibilities between the clients and the service providers to determine who is clearly responsible for securing what in the cloud, depending on the service model being deployed. To achieve a clarification, the Cloud Special Interest Group of the Security Standards Council promulgated a responsibility delineation matrix. This has been slightly modified to include programming tools as shown in figure 1 below.

IaaS	PaaS	SaaS	
Interface	Interface	Interface	Key  
Application	Application	Application	
Programming Tool	Programming Tool	Programming Tool	
Operating System	Operating System	Operating System	
Hypervisor	Hypervisor	Hypervisor	
Compute/Storage	Compute/Storage	Compute/Storage	
Network	Network	Network	
Building Facility	Building Facility	Building Facility	

Different adaptive models or proven traditional data security mechanisms ranging from cryptography to steganography have been deployed to secure data in the cloud.

Cryptography is the practice of secret writing, or more precisely, of storing information in an encrypted form which allows it to be revealed only to intended recipients after decryption. A cryptosystem is a method to accomplish cryptography. Cryptanalysis is the practice of circumventing or cracking such attempts to hide information. Cryptology includes both cryptography and cryptanalysis (Nikos and Eleftherios, 2000). Cryptographic implementation schemes and standards have evolved over the years. Early cryptographic schemes/algorithms were standardized by the National Bureau of Standards of the U.S in 1973 to constitute the Data Encryption Standards (DES). These early cryptographic algorithms include: Rivest Shamir and Adelman (RSA) algorithm; Diffie-Hellman algorithm; and CAST-256 algorithm. Concerted effort to improve cryptographic algorithm standards resulted in the development of the Advanced Encryption Standards (AES) in the standardization process christened ‘Encryption Olympics’. This was actually a five year contest which attracted experts in the cryptographic community. It was initiated in 1977 and a winner was announced in November, 2001.

NIST maintained that the evaluation criteria were separated in three categories:

- Security: namely actual security, random permutation properties and mathematical basis.
- Cost: i.e. computational efficiency and memory requirements (software & hardware)
- Algorithm and implementation characteristics: flexibility, hardware & software suitability and simplicity of design.

The contest culminated in the selection of a total of fifteen designs for evaluation, out of which five algorithm designs emerged finalists. These algorithms were namely, Rijndael, Serpent, Twofish, RC6 and MARS. After a rigorous selection process, Rijndael emerged the overall winner and Serpent was the first runner up. Interestingly, Serpent was adjudged to be the most secure of all the finalists but only lagged behind Rijndael in terms of the faster implementation speed of the latter which was attributed to its fewer rounds. Serpent's implementation speed was adjudged to be satisfactory.

Steganography is the practice of covering/hiding a message in such a way that no one else except the intended recipient knows of the existence of the message. Zadiraka and Kudin (2013) explained that steganography is the process of hiding one medium of communication (text, sound or image) within another that is situated in a separate file. Steganography involves encoding secret information in such a way that the very existence of the information is concealed under the image, sound or picture where it is hidden. The image/sound/video that the underlying message is hidden in is referred to as a carrier or cover file or signal. Garima and Naveem (2014) posited that the main advantage of steganography over cryptography is that secret message does not attract attention to itself as message can be concealed under image file, video file etc. They also posited that steganography is an efficient data hiding approach that comes handy when encryption is not permitted.

Distributed Steganography is an extension of the steganography model which aims to strengthen the data hiding concept of steganography. It involves fragmenting the message and hiding it in various carrier/cover files making the detection of the entire message extremely difficult, approaching impossibility. This research proposes the combination of distributed steganography and a highly secure variant of cryptography referred to as serpent algorithm. Consequently, these techniques are expounded in the succeeding sub-sections.

2.1.1 Serpent Cryptographic Algorithm.

Serpent algorithm was designed to provide users with the highest practical level of assurance that no shortcut attack will be found. To achieve this, its design was limited to well understood mechanisms, with a view to relying on the existing experience of block cipher cryptanalysis. Serpent was designed with twice as many rounds as are sufficient to block all known shortcut attacks.

Serpent is a symmetric key block cipher algorithm with a block size of 128 bits and supports a key size of 128, 192 or 256 bits (Anderson et al., 1998). The cipher is a 32-round substitution-permutation network operating on a block of four 32-bit words. Each round applies one of eight 4-bit to 4-bit S-boxes 32 times in parallel. Serpent was designed so that all operations can be executed in parallel, using 32 bit slices. This maximizes parallelism, but also allows use of the extensive cryptanalysis work performed on DES.

Serpent took a conservative approach to security, opting for a large security margin: the designers deemed 16 rounds to be sufficient against known types of attack, but specified 32 rounds as insurance against future discoveries in cryptanalysis.

The Serpent cipher is in open source. There are no restrictions or encumbrances whatsoever regarding its use. As a result, anyone is free to incorporate Serpent in their software (or hardware implementations) without paying license fees.

2.1.2. Features of Serpent Algorithm Specification.

Billham et al. (1998) exhaustively analyzed the serpent cipher and highlighted some salient features which Serpent designers took into consideration. These features include:

1) A simple block cipher that is easy to analyze and implement. The Serpent algorithm is open source and also easily implementable.

2) Serpent's block cipher has more rounds than are needed to block futuristic attacks.

Improvements in cryptanalysis usually increase the number of rounds required.

3) It's block cipher uses only well understood primitives. S-boxes and SP-networks have been around for over a quarter of a century, so it is less likely that surprising new attacks will be found on them. Serpent was designed with these considerations taken into cognizance.

4) It is so simple that it can be optimized in high level languages such as C, Ada, Java and Python. So a developer can avoid many of the errors that creep into assembly language routines.

5) The three most important aspects of algorithm performance are hardware complexity, software speed and memory cost. Serpent does extremely well with these criteria.

6) The Serpent ciphers were inspired by recent ideas for bitslice implementation of ciphers. However, unlike the bitslice implementation of DES, which encrypts 64 different blocks in parallel in order to gain extra speed, Serpent is designed to allow a single block to be encrypted efficiently by bit slicing. This allows the usual modes of operations to be used, so there is no need to change the environment to gain the extra speed (Billham, (1997).

7) Serpent achieves its high performance by a design that makes very efficient use of parallelism.

2.2 Distributed Steganography.

Steganography refers to any methodology used to hide a message (including text, sound, or picture) in a separate file. The most common method is to use the least significant bits of an image to store data. For example, in a high resolution graphics file, each pixel is represented by 24 bits. By using the least significant (i.e. the last 1 or 2 bits) to store other data, the image is not compromised and data is hidden in the image.

William (2013) maintained that there have been adaptations to the underlying technique, These adaptations include :

1) **Spread Spectrum Steganography:** This is primarily concerned with hiding an image/text within another image so that errors are minimal and detection of the image is more difficult.

2) **Video Steganography:** The purpose of this innovation is to hide some signal in a video transmission.

3) **Audio or Video Steganography.** In this adaptation, two signals (the message and the carrier) are combined to form a new signal. This invention also includes calibration data to facilitate adding and retrieving the hidden signal.

4) **Encryption Based Selection System for Steganography.** The aim of this modification is to integrate encryption with steganography. This research is premised on this adaptation.

Distributed steganography is an enhancement of classical or traditional steganography which is particularly concerned with how to fragment the message and hide it in various carrier/cover files making the detection of the entire message extremely difficult, if not entirely impossible. In this process, the message is distributed across multiple carrier signals/sources in order to further hide the message. For example, a single text message would be broken into blocks, each block hidden in a different image. Another aspect of this process is that the block size can vary and the blocks are not necessarily stored in order. This means that the first carrier file will not necessarily hold the first segment of the hidden message/file. This is applying permutation to the blocks. It should be noted that many cryptographic algorithms employ permutation along with substitution in order to encrypt files.

William (2013) illustrated the distributed steganography process using a smaller block size. He considered an example using 8 bit blocks on a message “Steganography is cool”. Each character represents 8 bits, so every 8 characters would be a separate block. Recalling that blanks are also represented by 8 bits, so this message would have 5 separate blocks stored in 5 separate images. A brief overview of the process of distributed steganography is shown in figure 2 below.

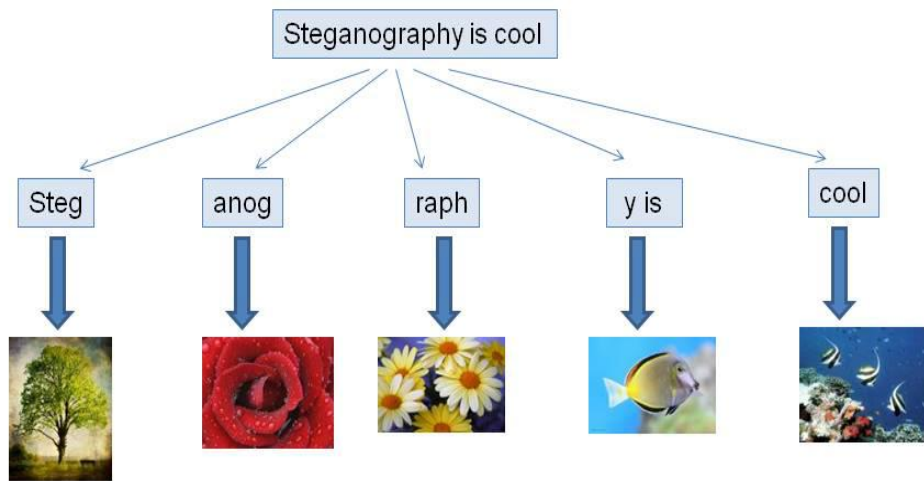


Fig. 2: On Overview of Distributed Steganography (William, 2013).

The challenge this process needs to overcome is how to retrieve the blocks. This issue would involve knowing how many blocks total were to be retrieved, the order of each block (i.e. is this block 2 of 4, 3 of 7, etc.), and knowing the carrier/cover file to retrieve the blocks from. This process deals with all three issues.

Each block stored in an image would have an additional 2 bytes (16 bits) appended to the image. The first byte would contain information to specify which block this was (i.e. block 3 of 9), and the second byte would store the total number of blocks the message contained (i.e. 9 blocks). Since 8 bits can store decimal numbers between 0 and 255 this would necessitate breaking a message down into no more than 255 blocks. The size of the block would be determined by the size of the original message divided by 255.

2.2.1 Review of Implementation Approaches of Distributed Steganography.

Distributed Steganography concept could be implemented in several adaptive approaches (William, 2013). These include:

(1) Use of Block pointers.

In this implementation approach, Additional bytes may be added to the hidden message to indicate block number and total blocks. These additional bytes are called block pointers. For example one could use 2 bytes (16 bits) to store the value of the current block and an additional 2 bytes (16 bits) to store the total number of blocks. This would allow a message to be broken into 65,535 total blocks. Use of up to 4 bytes (64 bits) for the value of the current block and 4 bytes (64 bits) for the total number of blocks would allow a message to be broken into 4,294,967,295 blocks. This would be appropriate for video or audio messages hidden in audio or video signals. The use of block numbering is similar to how TCP packets are sent over a network. Each packet has a number such as 'packet 2 of 10'. This same methodology is applied to hiding blocks of data in diverse images. This requires distributed steganography to have a key, much like the keys used in encryption. However this key would contain the following information:

i). Block Size

ii). Size of block pointer (i.e. the bytes used to indicate block numbering)

The preferred way to find the location of the images containing secret messages would be to add that information to the key. This information could be an IP address or URL to find the image at (if images are stored at different locations), or the image name (if all images are on a single storage device). Notice that it is possible to store images on web pages, file servers, or FTP servers. This means the actual message could be fragmented and stored around the internet in various locations. In some cases, it could even be stored on third party servers without their knowledge.

(2) Using Pre-determined Locations

In this steganography implementation process, the locations would be pre-determined. For example, messages would always be hidden in specific images at pre-determined locations. Thus the person who needs to receive those messages would simply check those images at regular intervals.

In another adaptation of this process, rather than embed the message into an image, it could be embedded into audio or video formats. The only alteration required would be the location of the carrier image would instead be the location of a video or sound file (.mp3, .wave, etc.). The actual encoding of the message could be done with any standard steganography technique, such as using the least significant bits to store the hidden message. It is advisable to have the message first encrypted using any preferred encryption algorithm, before hiding it using distributed steganography.

(3) Using Diverse Media Types.

In this approach a single message would be distributed in diverse media. This means some blocks would be embedded into images (.jpg, .bmp, etc.), others embedded into sound files (.wav, .mp3) and yet others could be embedded into video (.mov, .avi). This approach requires the decoding software to accommodate diverse media.

3.0 Related Work

Cryptography and Steganography share common objectives of ensuring adequate protection of data. These techniques have witnessed extensive research and have been tested and proven to be effective. Several approaches to securing data in the cloud have been documented in literature.

Mrinal and Trijit (2014) examined the problem of security in cloud computing and proposed an effective and efficient steganographic approach for enhancing security on data-at-rest in cloud data storage centers.

Abikoye et al. (2012) developed a system that combined the cryptography and steganography techniques to provide an efficient system of hiding data from authorized users. They employed an audio medium for steganography and the Least Significant Bit algorithm to encode the message in the audio file.

Mohammad and Abdelfatah (2010) proposed an approach that described two steps for hiding secret information by using the public steganography based on matching method. The first step, determines the shared stego-key between the two communication parties over the networks by applying Diffie Hellman Key exchange protocol. The second step in the proposed method is ensures that, the sender uses the secret stego-key to select pixels that it will be used to conceal. Each selected pixel is then used to hide 8 bits binary information depending on the matching method.

Demchenko et al.(2013) presented and developed the Inter-cloud Architecture that handles problems with multi-domain heterogeneous cloud based applications, integration, inter-provider and inter-platform interoperability. They evaluated the security issues in provisioning complex heterogeneous multi-provider intercloud infrastructures. Their investigation provided veritable basis for further inter-cloud security infrastructure development.

Sharon et al. (2013) explored the vulnerabilities and threats of cloud storage. Cloud storage characterizes one of the domains of cloud computing that affects the different cloud service models.

Garima and Naveen (2014) proposed an approach for securing data in the cloud using Digital Signature Algorithm, Data Encryption Standard and Steganography. Their aim was to combine these three algorithms to provide maximum security, authenticity and data integrity in the cloud.

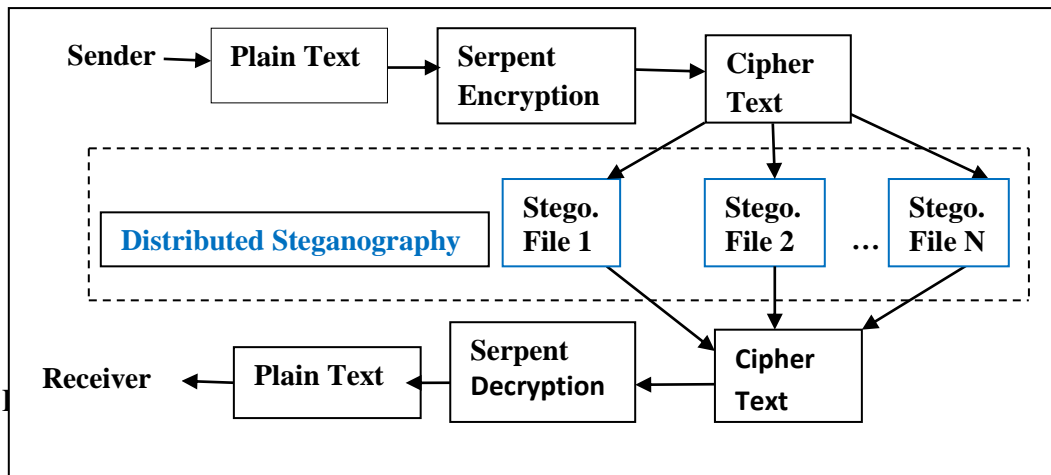
Yuri et al. (2012) developed and presented the architectural framework for cloud based infrastructure services provisioning. They proposed architecture, which intended to provide a basis for building multilayer cloud services integration framework and to allow optimized provisioning of computing, storage and networking resources. They also proposed Inter-Cloud architecture, which would facilitate cloud services' interoperability and integration.

Selvn et al (2013) discussed the issue of securing data while storing it in the cloud server. They suggested following few steps, including the implementation of: (1) new data displacement strategies, (2) service level agreement between the user and the cloud service provider and (3) quality of service verification.

Ravij et al. (2013) proposed a technique, which is actually a combination of Identity Based Encryption (IBE) and Mediated RSA (mRSA) techniques for ensuring secure Cloud environment.

4.0 The Proposed Approach.

In the proposed approach, Serpent encryption algorithm is combined with distributed steganography to provide an enhanced layer of protection for data in the cloud. The proposed approach aims to amplify the strength of steganography by employing distributed steganography. In this approach, serpent algorithm, the most secure of the Advanced Encryption Standard algorithms, is first applied by the message sender for encryption and then its resulting cipher text is subjected to distributed steganography. On the receiving side, the distributed steganography files are recombined by reversing the distributed steganography and the resulting cipher text is then decrypted by reversing the serpent algorithm. Figure 3 below depicts the design of the proposed approach.



4.0.1 Justification of the Proposed Approach.

The following salient points justify the significance and practicability of this proposed approach.

- Serpent encryption algorithm has been critically evaluated. With satisfactory implementation speed, Serpent algorithm was adjudged to be the most secure of the Advanced Encryption Standards in the NIST organized ‘encryption olympic’ contest.
- Serpent cipher is open source and so its codes are readily available at no cost.
- Serpent cipher codes are easily implemented and can be optimized using several programming languages including Java, and Python.
- Steganography technique has been tested and proven that its strength can be amplified by integrating it with cryptography.

5.0 Conclusion and Future Work.

In this paper, a unified approach for ensuring data confidentiality and integrity was proposed. The proposed approach is essentially a strategic combination two proven techniques, serpent encryption algorithm and distributed steganography, to achieve a robust data security mechanism. The idea of extending this approach by combining a homomorphic variant of the serpent encryption algorithm with steganography would present viable grounds for future research.

Acknowledgement

Dr. C. Ugwu, the Head of Computer Science Department, University of Port Harcourt, deserves special acknowledgement for providing invaluable guidance in facilitating the completion of this research.

References

- Alshuwaier, F. A., Alshwaier, A. A., & Areshey, A. M. (2012). Applications of Cloud Computing in Education. In *8th International Conference on Computing and Networking Technology*. 26–33.
- Anderson, R.J., Biham, E., Knudsen, L.R (1998). Serpent: A Proposal for the Advanced Encryption Standard. *Proceedings of the AES conference*. Springer, USA
- Biham, E. (1997). A Fast New DES Implementation in Software. *Fast Software Encryption. 4th International Workshop, FSE '97*, Springer. pp 260-271
- Demchenko, Y., Makkes, M., Strijkers, R., Ngo, C. and Laat, C. (2013). Intercloud Architecture Framework for Heterogeneous Multi-Provider Cloud based Infrastructure Services Provisioning. *The International Journal of Next-Generation Computing (IJNGC)*, 4(2).
- Garima S, and Naveen S (2014) . Triple Security of Data in Cloud computing. *International Journal of Computer Science and Information Technologies*. 5(4)
- Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. Retrieved from <http://csrc.nist.gov/publications/nistpubs/800-145/SP800145.pdf>
- Messer E. (2012). Gartner: Network Virtualization will lead to security control changes. Network Work, Retrieved <http://www.network.com/news/2012/061212-gartner-macdonal260107.html>
- Mohammad, A. A., and Abdelfatah, A. Y. (2010). Public-Key Steganography Based on Matching Method. *European Journal of scientific Research*. 4(2).
- Nikos Moshopoulos and Eleftherios Chaniotakis (2000). *A Survey of Cryptography Algorithms*: Unpublished Thesis, university of Athens, Greece..
- Ravij, K., Nishant, S . and Sutaria, K. (2013). Ameliorate Security Policy Using Mediated RSA and Identity Based Cryptography in Cloud Computing. *Journal of information, knowledge and research in computer engineering*, 2, ISSN: 0975 – 6760, pp. 389.
- Selvn, M., Subbiah, S. and Ramkumar, T. (2013). Enhanced Survey and Proposal to Secure the Data in Cloud Computing Environment., *International Journal of Engineering Science and Technology (IJEST)*. 5(1).
- Sharon, I., Kumar, C., Andrew, W., and Jeevakumar, J. (2013) “A Survey on Security Threats and Vulnerabilities in Cloud Computing”, *International Journal of Scientific and Engineering Research*. 4(3).

Taniya (2013). Introduction to Cloud Security. *International Journal of Electronics and Communication Engineering and Technology (IJECEET)*. pp252-260

William Charles Easttom (2013). Method and Apparatus of Performing Distributed Steganography of A Data Message. *Open Invention Network*, LLC. North Carolina, USA.

Yuri, D., Ngo, C., Makkes, X., Strijkers, R., Laat, C. (2012). Defining Inter-Cloud Architecture for Interoperability and Integration. University of Amsterdam, System and Network Engineering Group, Cloud Computing, France.

Zadiraka V.K., and Kudin A. M. (2013). Cloud Computing in Cryptography and Steganography. *Cybernetics and Systems Analysis*. 49(4).