Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 85 (2016) 535 – 542

Procedia Computer Science

International Conference on Computational Modeling and Security (CMS 2016)

Security Algorithms for Cloud Computing

Akashdeep Bhardwaj[a*], GVB Subrahmanyam[b], Vinay Avasthi[c], Hanumat Sastry[d]

[a]*University of Petroleum and Energy Studies, Bidholi Dehradun 248001, India*
[b]*Tech Mahindra, Infocity, Hyderabad 500081 India*
[c]*University of Petroleum and Energy Studies, Bidholi Dehradun 248001, India*
[d]*University of Petroleum and Energy Studies, Bidholi Dehradun 248001, India*

Abstract

With growing awareness and concerns regards to Cloud Computing and Information Security, there is growing awareness and usage of Security Algorithms into data systems and processes. This paper presents a brief overview and comparison of Cryptographic algorithms, with an emphasis on Symmetric algorithms which should be used for Cloud based applications and services that require data and link encryption. In this paper we review Symmetric and Asymmetric algorithms with emphasis on Symmetric Algorithms for security consideration on which one should be used for Cloud based applications and services that require data and link encryption.

Peer-review under responsibility of the Organizing Committee of CMS 2016

Keywords: Cryptography, Security Algorithm, Symmetric, Asymmetric, RSA, RC6, AES, 3DES, MD5

1. Introduction

Imagine two people who share critical secret information have to split up. This requires them to share and communicate their data and information from a distance, even as there lays a threat of an eavesdropper having the ability to stop, interfere or intercept their communications and seeks that same information. They decide to lock their information in a box using a lock that only the other knows the combination to and has the key to open it. The box is locked and sent over to the other user who uses the combination key to unlock the box and read its contents. In simple terms, Cryptography [1] can be seen as a method of storing and disguising confidential data in a cryptic form so that only those for whom it is intended can read it and are able to communicate information in the presence

* Corresponding author. Tel.: +91-987-327-6660.
E-mail address:Bhrdwh@yahoo.com

Peer-review under responsibility of the Organizing Committee of CMS 2016
doi:10.1016/j.procs.2016.05.215

of an adversary and the security algorithms mitigate security issues by use of cryptography, authentication and distributing keys securely. Cryptography is thus the science of making data and messages secure by converting the end user data to be sent into cryptic non-readable form and encrypting or scrambling the plaintext by taking user data or that referred to as clear text and converting it into cipher text [2] and then performing decryption which is reverting back to the original plain text. With this ability, Cryptography is used for providing the following security:

- Data Integrity: information has value only if it is correct, this refers to maintaining and assuring the accuracy and consistency of data, its implementation for computer systems that store use data, processes, or retrieve that data.
- Authentication for determining whether someone or something is, in fact, who or what it is declared to be.
- Non Repudiation: is the assurance that a party, contract or someone cannot deny the authenticity of their signature and sending a message that they originated.
- Confidentiality: relates to loss of privacy, unauthorized access to information and identity theft.

Fig 1: Encryption and Decryption process

In pure science terms [3], Cryptography is the science of using mathematics for making plain text information (P) into an unreadable cipher text (C) format called encryption and reconverting that cipher text back to plain text called as decryption with the set of Cryptographic Algorithms (E) using encryption keys (k1 and k2) and the decryption algorithm (D) that reverses and produces the original plain text back from the cipher text. This can be interpreted as Cipher text C = E {P, Key} and Plain text C = D {C, Key}

Defining some terms used in Cryptography:

- Plaintext is the original intelligible source information or data that is input to algorithms
- Cipher text is the scrambled message output as random stream of unintelligible data
- Encryption Algorithm substitutes and performs permutations on plain text to cipher text
- Decryption Algorithm is encryption run in reverse by taking the secret key and transforming the cipher text to produce the original plain text
- Keys are used as input for encryption or decryption and determines the transformation
- Sender and Recipients are persons who are communication and sharing the plaintext

With respect to Cloud computing, the security concerns [4] are end user data security, network traffic, file systems, and host machine security which cryptography can resolve to some extent and thus helps organizations in their reluctant acceptance of Cloud Computing. There are various security issues that arise in the Cloud:

- Ensuring Secure Data Transfer: In a Cloud environment, the physical location and reach are not under end user control of where the resources are hosted.
- Ensuring Secure Interface: integrity of information during transfer, storage and retrieval needs to be ensured over the unsecure internet.
- Have Separation of data: privacy issues arise when personal data is accessed by Cloud providers or boundaries between personal and corporate data do not have clearly defined policies.
- Secure Stored Data: question mark on controlling the encryption and decryption by either the end user or the Cloud Service provider.
- User Access Control: for web based transactions (PCI DSS), web data logs need to be provided to compliance auditors and security managers.

Security Algorithms are classified broadly as:

- Private Key / Symmetric Algorithms: Use single secret key are used for encrypting large amount of data and are have fast processing speed. These algorithms use a single secret key that is known to the sender and receiver. RC6, 3DES, Blowfish, 3DES are some prime examples of this algorithms.

- Public Key / Asymmetric Algorithms: Use a key pair for cryptographic process, with public key for encryption and private for decryption. These algorithms have a high computational cost and thus slow speed if compared to the single key symmetric algorithms. RSA and Diffie Hellman are some types of public key algorithms.
- Signature Algorithms: Used to sign and authenticate use data are single key based. Examples include: RSA, DH
- Hash Algorithms: Compress data for signing to standard fixed size. Examples include: MD5, SHA
- Some other ways of classifying Algorithms based on their processing features as below

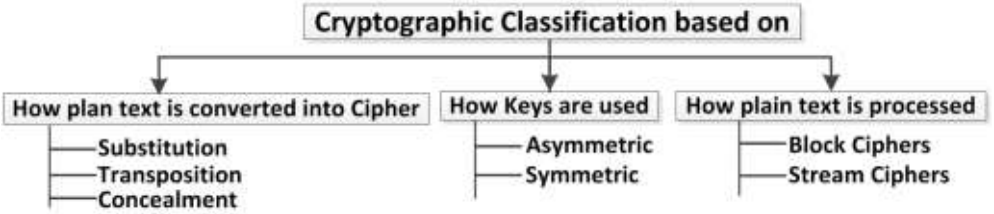

Fig 2: Classification of Algorithms

With several Cloud services, Servers and hosted applications under IT management, most Cloud providers have no defined process to ensure security of data from threats and attacks [5]. Cyberattack these target the end user data for which the Cloud Service providers seek to try and secure by using Cryptographic algorithms whose primary goal is to make it as difficult as possible to ensure decrypting the generated cipher text from the plain text. When the key length is long, that makes it harder to decrypt the cipher texts, which in turn make the algorithms efficient and effective.

2. Asymmetric Algorithms

Asymmetric Algorithms [6] a pair of related key, one key for encryption called the Public key and a different but inter related key for Decryption called the Private keys when performing transformation of plain text into cipher text. The main asymmetric algorithms are ECC, Diffie-Hellman and RSA.

2.1 RSA:

RSA Algorithm named after its inventers (Rivest, Shamir, and Adelman) is best suited for data traveling to/from Web and Cloud based environments. In working with Cloud Computing, the end user data is first encrypted and then stored on the Cloud. When the data is required, the end user simply needs to place a request to the Cloud Service provider for accessing the data. For this the Cloud service provider first authenticates the user to be the authentic owner and then delivers the data to the requester using RSA Asymmetric Algorithm. This algorithm has support from .NET Security Framework as well.

Here two keys involved – first the Public Key [7] which known to all and the other Private Key which is known only to the end user. Data conversion from plain text to cipher text is done using Public Key by the Cloud service provider and the cipher text to plain text decryption is done by the end user using Private Key as the Cloud service consumer. Once the user data is encrypted with the Public Key, that cipher data can only be decrypted with the corresponding Private Key only. In this Algorithm, prime numbers are used to generate the public and private keys based on mathematical formulas and by multiplying the numbers together. This uses the block size data in which plain text or the cipher texts are integers between 0 and 1 for some n values. Here the processed plaintext is also encrypted in blocks and the binary value of each block needs to be less than the number (n). RSA being multiplicative homomorphic which essentially means that to find the product of the plain text, multiply the cipher texts so that the outcome of the result is the cipher text of the product.

2.2 Diffie-Hellman Key Exchange (D-H):

This is a method for exchanging cryptographic keys [8] by first establishing a shared secret key to use for the inter communication and not for encryption or decryption. This key exchange process ensures the two parties that have no prior knowledge of each other to jointly establish a shared secret key over unsecure internet.

Transformations of keys are interchanged and both end up with the same session key that looks like a secret key. Then each can then calculate a third session key that cannot easily be derived by an attacker who knows both exchanged values. This key encrypts the subsequent communications using a symmetric key cipher but is vulnerable to the Man-in-the Middle (MITM) attack. This key exchange is not used for exchanging real large data unlike RSA.

3. Symmetric Algorithms

Symmetric algorithms involve a single shared secret key [9] to encrypt as well as decrypt data and are capable of processing large amount of data and from computing standpoint are not very power intensive, so has lower overhead on the systems and have high speed for performing encryption and decryption. Symmetric algorithms encrypt plaintexts as Stream ciphers bit by bit at a time [10] or as Block ciphers on fixed number of 64-bit units.

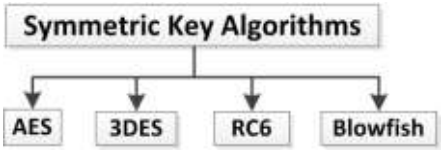

Fig 3: Symmetric Algorithms

There are however few problems with Symmetric Algorithms:

- Exchanging Shared Secret Key [11] over unsecure internet.
  Symmetric-key algorithms share secret keys required by the sender and receiver during encryption or decryption process. In case a third person gains access to the secure secret key, cipher text messages can easily be decrypted. The fact of having one single secret key algorithm is the most critical issue faced by Cloud service providers when dealing with end users who communicate over unsecure internet. The only option is to have that secret key be changed often or kept as secure as possible during the distribution phase.
- Problem confirming if the content is altered or actually sent by the claimed sender.
  If a hacker has the secret key, decrypting the cipher text, modifying the information being sent with that key and send to the receiver. Since a single key is involved during the crypto process, either side of the transactions can get compromised. Such data integrity and non-repudiation issues however need to involve the use of Digital signatures or Hashing functions like MD5.
- Tools for cracking Symmetric encryption
  By use of Brute force [12] by running hacking tools that have the ability crack the combinations and keys to determine the plaintext message and perform Cryptanalysis where the attacks are focused on the characteristics of the algorithm to deduce a specific plaintext or the secret key. Then hackers are able to figure out the plaintext for messages that would use this compromised setup.

4. Related Work Performed

With DDoS and Malware attacks on the rise, Cloud Providers are giving more focus on having end user data as secure as possible and having low priority for cloud performance due to inconsistent selection of algorithms for encryption and encoding. By selecting the right cryptographic scheme end user data security can be achieved without losing out on cloud performance. Since Algorithm analysis is an essential in gathering the knowledge against any accidental or unintentional use algorithm that may prove to be inefficient or significantly impact application system performance due to encryption or decryption. For those cloud based web applications or portals needing real time or time sensitive data, an algorithm that might be taking a long time to long to run would prove a hindrance for the real time application as it may render the results to be useless. Such in efficient algorithm might end up needing lots of computing power or storage to execute over the cloud, making the algorithm useless in that

environment.

4.1. Comparison Parameters

Authors compared Symmetric encryption algorithms and encoding algorithms using size and time to decide on selection of the right algorithms based on the parameters as

- File Size: indicates file of different size to be taken
- Encryption Computation Time: time an algorithm takes to produces a cipher text from a plain text
- Encoding Computation Time: time taken by encoding algorithm to produce a hash code

Performance metrics were collected based on the following:

- Encryption & Decryption Time: This is calculated as the time required for encryption which involves converting the plain text payload file into cipher text. The authors used the encryption time to find the through put which indicated the computation cost i.e. the encryption speed. The decryption time is calculated for the amount of time required for converting the cipher text back into the plain text.
- CPU Processing Time: This is determined as the time CPU is committed for the process and reflects the CPU load during the encryption process. The CPU Clock Cycle and Battery power are the energy consumed during encryption and decryption process.
- Size of payload to be tested: This is actual size of the test file that is being used for the experimental work.

The authors then used the below infrastructure for our data gathering research work:

- Connectivity: 1Mbps WAN circuit link connected to a public Cloud server provider
- Cloud Simulation: Hosted Web application server on the IaaS systems for cloud environment
- Working environment:
    - Programming language environment - Java
    - Setup one 64 bit Windows Server 2008 Operating system
    - Running on VMware based a Virtual machine
    - Over hardware as Intel Core i5-3230M CPU @ 2.66GHz, 8GB memory.

4.2 Performance Evaluation

The below mentioned actions were performed as input using different algorithms to encrypt the data (text file) to determine the time required for reading the file, encrypting it, creating the encrypted data, then sending the data to a cloud location and receiving a confirmation.

Fig 4: Encrypting text file to send to the cloud

The input variables are

- File upload: D:\SACC\Data\Encrypt.txt (input)
- Choosing algorithm:
- Encoding Hash
- Key size:
- Mode: Encrypt or Decrypt

5. Performance Results

The data from experimental work on Symmetric algorithms is depicted below by using varied file sizes as input and recording the computation cost for those algorithms. Encoding algorithms checks for data integrity for end user data on the cloud and computation cost data obtained for different algorithms by varying the size of payload.

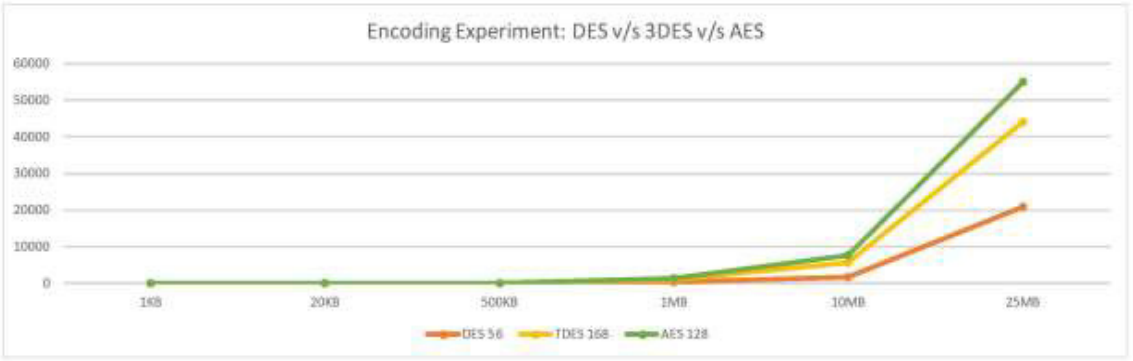

Fig 5: Computational Cost for Encryption

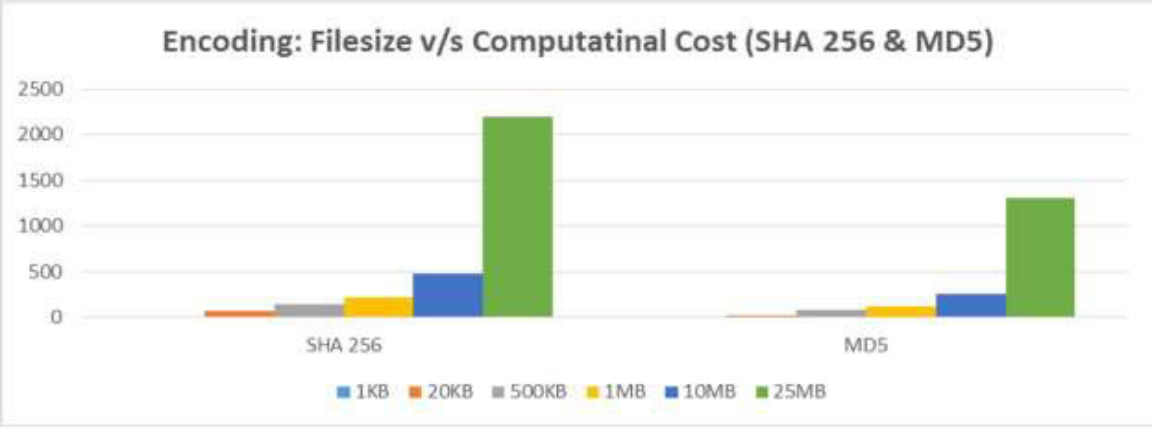

Fig 6: Computational Cost for Decryption

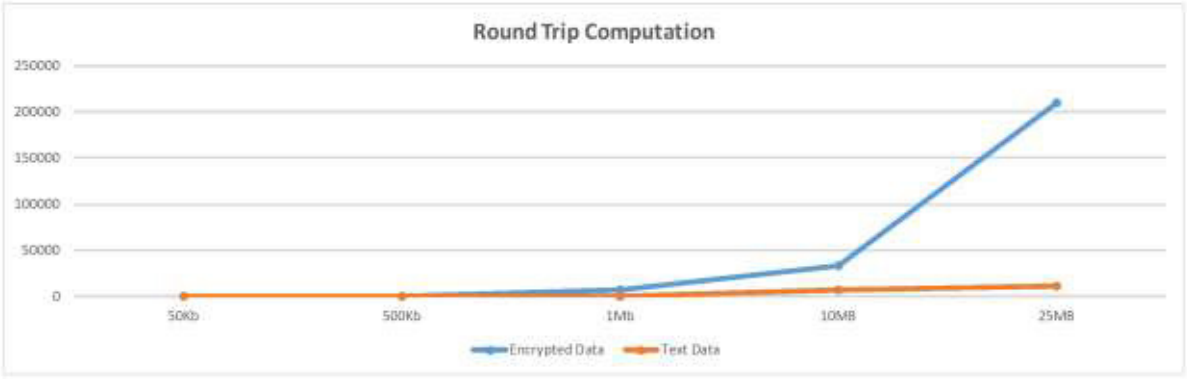

Fig 7: Execution Time (Text file to encrypted file and back)

Further observations from the work performed:

- Data Security for Cloud based applications can be increased by using RSA and AES Encryption algorithms
- When using keys as 1024 bit RSA and 128 bit AES, determining the private key is not possible even if the attacker has the public keys generated
- After the end user logins to the Cloud web portal, accesses the applications but does not log out and in fact just leaves the session idle, then in this case if an attacker breaks in to the user system attempting to download and access the data from the user system, then the attacker would be required to enter the private key.
- In case the attacker in his attempt to break in to the user system is successful, even able to somehow guess the private key and then go on to download the encrypted data.
- The attacker might be successful in getting the encrypted data but still accessing the original data might still not be possible.

6. Conclusions

With Cloud computing emerging as a new in thing in technology industry, public and private enterprise and corporate organizations are either using the Cloud services or in process of moving there but face security, privacy and data theft issues. This makes Cloud security a must to break the acceptance hindrance of the cloud environment. Use of security algorithms and ensuring these are implemented for cloud and needs to be properly utilized in order to ensure end user security. The authors analyzed Symmetric algorithms for different encryption and encoding techniques, found AES to be a good candidate for key encryption and MD5 being faster when encoding.

References

1. Leena Khanna, Anant Jaiswal, “Cloud Computing: Security Issues and Description of Encryption Based Algorithms to Overcome Them”, IJARCSSE 2013
2. G Devi, Pramod Kumar “Cloud Computing: A CRM Service Based on a Separate Encryption and Decryption using Blowfish algorithm” IJCTT 2012
3. Simarjeet Kaur “Cryptography and Encryption in Cloud Computing”, VSRD International Journal of CS and IT, 2012
4. Nelson Gonzalez, Charles Miers, Fernando Redigolo, Marcos Simplicio, Tereza Carvalho, Mats Naslund, Makan Pourzandi “A quantitative analysis of current security concerns and solutions for cloud computing”, Springer 2012.
5. Ronald Krutz, Russell Vines, “Cloud Security: A Comprehensive Guide to Secure Cloud Computing” Wiley Publishing 2010
6. Behrouz Forouzan, “Cryptography and Network Security”, McGraw-Hill Special Indian Edition 2007
7. Wayne Jansen, Timothy Grance “Guidelines on Security and Privacy in Public Cloud Computing”, National Institute of Standards and Technology 2011
8. Akhil Behl “Emerging Security Challenges in Cloud Computing”, IEEE 2011
9. Maha Tebba, Saïd Haji Abdellatif Ghazi, “Homomorphic Encryption Applied to the Cloud Computing Security”, World Congress on Engineering 2012
10. Cloud Security Alliance (CSA), “Security Guidance for critical Areas of Focus in cloud computing V3.0” CSA 2015
11. Ayan Mahalanobis, “Diffie-Hellman Key Exchange Protocol, Its Generalization and Nilpotent Groups.” 2005
12. Neha Jain, Gurpreet Kaur, ‘Implementing DES Algorithm in Cloud for Data Security”, VSRD International Journal of CS and IT, 2012

13. Mandeep Kaur, Manish Mahajan, “Implementing Various Encryption Algorithms to Enhance The Data Security Of Cloud In Cloud Computing” VSRD International Journal of Computer Science & Information Technology 2012
14. Jeeva, Dr. Palanisamy, Kanagaram, “Comparative Analysis of Performance Efficiency and Security Measures of some Encryption Algorithms”, IJERA ISSN: 2248-9622 Vol. 2, Issue 3, 2012
15. Dr. Sarbari Gupta,”Securely management crypgraphic keys used within acloud environment”, NIST Cryptographic Key management workshop, 2012
16. Dr. R. Chandramouli “Key Management Issues in the Cloud Infrastructure”, Workshop on Cloud Computing, 2013
17. Sandro Rafaeli, “Survey of key management for secure communication”, ACM Computing Surveys, 2013
18. S. Anahita Mortazavi, Alireza Nemaney Pour, Toshihiko Kato,”An Efficient Distributed Group Key Management using Hierarchical Approach with Diffie-Hellman and Symmetric Algorithm: DHSA”,CNDS Feb 2011
19. ENISA,”Algorithms, Key Sizes and Parameters Report,2013”, recommendations version 1.0 – October 2013
20. Y. Fan, L. Xiao-ping, D. Qing-kuan and L. Yan-ming, “A Dynamic Layering Scheme of Multicast Key Management,” IEEE 5th International Conference on Infomration Assurance and Security, Xian 2009
21. Rajesh Ingle, G. Sivakumar, "EGSI: TGKA based Security Architecture for Group Communication in Grid", 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 34-42, 17-20 May, 2010.
22. NIST, “Cloud Computing Synopsis and Recommendations”, Special publication 800-146, May 2012