ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Seccloud Protocol Implementation Using Aes Algorithm For Security And Privacy In Cloud Computing

Aneesha k Jose

P.G Student, Department of Computer Science and Engineering,
Karunya University, Coimbatore

Abstract: ***Cloud computing is 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. Data and computations can be moved to the huge data centers, called cloud. In Seccloud protocol data is encrypted and send to the cloud server and decryption takes place at the server before storage. Computations were also performed on this data at the cloud server. For encryption RSA algorithm was used initially. But it suffered from brute force attack and other mathematical attacks. RSA also consumed high encryption and decryption time while execution. In order to overcome this AES algorithm was introduced. The operations in AES are repeated for several stages called rounds. A change in a single bit in key or plain text results in a completely different cipher. This is an advantage over traditional stream ciphers. Also AES-256 is more secure than RSA-128 because it has 256-bit key - that means 2^{256} possible keys to brute force, as opposed to 2^{128} in RSA. Thus AES remains as the preferred algorithm for encryption and decryption in the cloud computing environment.***

Keywords: Seccloud, Storage Security, Computation Security, AES

1. INTRODUCTION

Security and privacy are the major challenges which inhibit the cloud computing wide acceptance in practice. Different from the traditional computing model in which users have full control of data storage and computation, cloud computing entails that the managements of physical data and machines are delegated to the cloud service providers while the users only retain some control over the virtual machines. Thus, the correctness of data storage and computation might be compromised due to the lack of the control of data security for data owners. We further classify cloud computing security into two major classes: They are Cloud Storage Security and Cloud Computation Security where the former is referred to ensuring the integrity of outsourced data stored at untrustworthy cloud servers while the latter refers to checking the correctness of the outsourced computation performed by untrustworthy cloud servers.

The correctness of data storage and computation might be compromised at the cloud due to the lack of the control of

data for data owners. Secure cloud computing focuses on the cloud storage security and cloud computation security. For that RSA algorithm was used initially to encrypt and decrypt the data stored at the cloud server. But it suffered from brute force and other mathematical errors. RSA also consumed high encryption and decryption time while execution. In order to overcome this AES algorithm was introduced which consumed less encryption and decryption time and also is not vulnerable to brute force attacks.

2. EXISTING METHODS

Safety of the data stored in the cloud servers has been compromised in many cases for monetary profit. Therefore it is essential to maintain the security and privacy of stored data and computation at cloud servers and therefore various methodologies were introduced.

2.1 Byzantine Fault Tolerant Algorithm (BFT)

BFT algorithm in (M. Castro E et al., 2008) ensures security of storage and computation at cloud servers. It is used in systems where there is no limitation of time. It can be used to replicate existing data. It is used to develop systems that do not fail under byzantine faults. If a fault is met it redoes computation till the desired output is met.

2.2 Remote data integrity checking protocol

This protocol in (Zhuo Hao., 2011) ensures security for stored data at cloud server. The remote data integrity checking protocol uses Homomorphic Verifiable Tags (HVT). A HVT of a message m checks whether the stored data is subject to modification or not. By using HVTs, the server can obtain a proof of possession for a set of file blocks that those file blocks were not modified at cloud server. Thus the client needs not have access to these file blocks to ensure storage correctness.

2.3 Distributed storage integrity auditing

This method allows the users to audit the cloud storage with less communication and computation cost as in (Cong Wang., 2009). It uses homomorphic token and distributed erasure coded data. Initially the files of users are distributed across different servers. Then compute tokens that cover a set of blocks and store those tokens at cloud user. When user wants to make sure of storage correctness he gives a set of block indices to Cloud Server. On receiving the challenge each Cloud Server computes a short signature over the specified blocks and returns them to user. The value of these signatures should match corresponding tokens precomputed by user. If they do not match it indicates that integrity of data stored at server is disturbed. Once error is detected, user asks server to send these erroneous blocks and corrects them. This creates an extra burden for the cloud user.

2.4 Robust Data Possession (RDP)

This method in (Reza Curtmola., 2008) integrates Forward Error Correcting codes (FEC) into Provable Data Possession (PDP). A file is first encoded using an FEC code to form an encoded file. Then PDP is applied on the encoded file instead of original file. Thus it has two benefits. Firstly it

prevents corruption of a large portion of file i.e. when there is a change in the blocks. This happens when the Cloud Server sells same storage space to multiple clients. Secondly it prevents corruption of a small portion of file i.e. changes within the block are also detected since FEC code is used.

2.5 Proofs Of Retrievability (POR)

POR in (A.Juels., 2007) helps a client to receive a proof from the server that its data is not deleted or modified at the server. POR encrypts file and randomly embeds a set of values called sentinels .Sentinels can be distinguished from file block. To verify storage correctness user or verifier gives positions of sentinels and asks to return the sentinel values. If the cloud server has modified or deleted a large portion of file, it cannot return the correct value of sentinel.

3. PROPOSED APPROACH

We consider a cloud computing model constituted of a number of cloud servers, S1,S2,...,SN, which are under the control of one or multiple cloud service providers (CSP). CSP allocates resources by means of customized Service Level Agreements. CSP divides such a large task into multiple small sub-tasks and allows them parallelly executed across up to hundreds of cloud servers. We assume the cloud user (CU), such as a mobile phone, a laptop, and an apple ipad, which has lower computation resource and smaller storage resource than those of the cloud servers. CU would submit storage service requests and computation service requests to CSP when it demands. Similar to existing secure storage auditing schemes, we also assume the existence of a number of verification agencies (VAs), which are chosen and trusted by CU and responsible for auditing the cloud services on data storage and computation.

3.1 The Basic Seccloud Protocol

The basic Seccloud protocol contains the following four phases.

- System Initialization

In this Module System Setup is done. The System Initialization Operator (SIO) generates the system parameters and master secret keys. When a cloud user applies for the cloud services, it first needs to register to SIO. The cloud user submits its identity ID to SIO and receives system parameters params and a secret key from SIO.

- Secure Storage

The next module is secure communication. Before the data are transmitted to the cloud, the cloud user first applies the storing space for its messages. To enable the storage data auditing, the cloud user needs to sign each transmission block to generate authenticated information. CU also chooses a trusted verification agency and then does signing operation to sign each block.

- Secure Computation

After receiving the packets, CSP first decrypts them by its own session key to obtain the data-signature pairs {D,U} and checks the signatures by verifying its secret key . When CSP receives the computation requests CSP divides such a large task

into multiple small sub-tasks and allows them executed parallelly on the different cloud servers based on the data positions. Each cloud server first finds the data in the position pi, computes the function.

- Computation Verification

To achieve the secure cloud computation, CUs need to perform result verification. VA not only checks if the data has been appropriately stored at the cloud server but also ensures if the computation process has been performed correctly. It is not possible to fetch each of the data block and re-compute each function to check if the cloud storage and computation is well performed. Thus, the probabilistic sampling technique is adopted here to reduce the overall verification cost.

3.2 Seccloud Protocol using RSA algorithm

RSA was the most commonly adopted public key cryptography algorithm. It can be used for key exchange, digital signatures and encryption of small blocks of data. The RSA algorithm involves the use of two types of keys. A public key which may be known by anybody and can be used to encrypt the message and a private key known only by the recipient and can be used to decrypt the message. The operation of RSA algorithm is as follows:

- Choose 2 distinct random prime numbers p and q
- Compute n = p*q
- Compute f(n) = (p-1)*(q-1)
- Choose an integer e, such that 1 < e < f (n) and gcd (e, f (n)) = 1
- Compute $d = e^{-1} \bmod [f(n)]$
- Publish the public encryption key: (e, n)
- Keep secret private decryption key: (d, n)

To encrypt a message the sender has to do the following:

- Obtain public key of recipient (e, n)
- Represent the message as an integer m in [0, n-1]
- Compute : $c = m^e \bmod n$

To decrypt the cipher text c the recipient does the following:

- Uses his private key (d, n)
- Computes: $m = c^d \bmod n$

When attempting to attack the RSA, the attacker has access to the public key (e, n) and wants the private key (d, n). There are three ways to attack RSA such as force the search of key, launch attacks during the decryption and mathematical attack. To get d: factorize n-> p and q -> f (n) -> d. For small numbers, it is very easy to hack a RSA code. The RSA algorithm is based on the fact that it is far more difficult to factorize a product of two primes than it is to multiply the two primes. Factoring n is the best known attack against RSA to date. Therefore it cannot guarantee 100% security, only added protection while storing data at the cloud server.

3.3 Seccloud Protocol using AES algorithm

AES is a symmetric block cipher with a block size of 128 bits. Key lengths can be 128 bits, 192 bits, or 256 bits called as AES-128, AES-192, and AES-256, respectively. AES-128

uses 10 rounds, AES-192 uses 12 rounds, and AES-256 uses 14 rounds.

Each round of AES performs the following functions:

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

The first three functions of an AES round are designed for cryptanalysis via the
method of confusion and diffusion. The fourth function actually encrypts the data. AES formats plaintext into 16 byte (128-bit) blocks, and treats each block as a 4x4 State array. It then performs four operations in each round.

- SubBytes

SubBytes add confusion by processing each byte through an S-Box. An S-Box is a substitution table, where one byte is substituted for another, based on substitution algorithm

- ShiftRows

ShiftRows provides diffusion by mixing data within rows. Row zero of the state is not shifted, row 1 is shifted 1 byte, row 2 is shifted 2 bytes, and row 3 is shifted 3 bytes.

- MixColumns

MixColumns also provides diffusion by mixing data within columns. The 4 bytes of each column in the State are treated as a 4-byte number and transformed to another 4-byte number via finite field mathematics

- AddRoundKey

The actual encryption is performed in the AddRoundKey function, when each byte in the state is XORed with the subkey. The subkey is derived from the key according to a key expansion schedule.

The overall AES encryption and decryption procedure in different rounds is shown in fig 3.1 below.

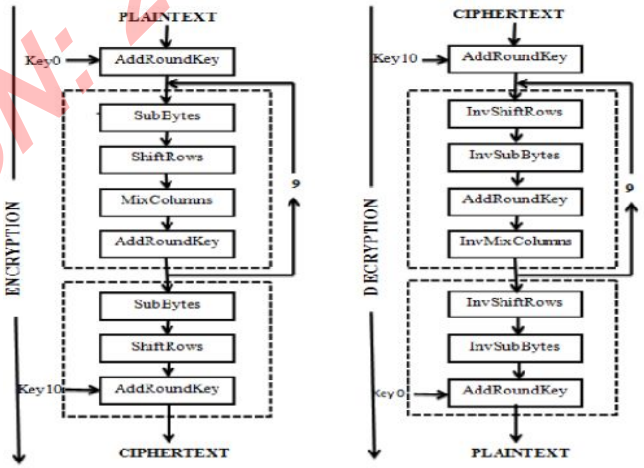

Fig 3.1- AES encryption and decryption

AES is a symmetric cryptographic algorithm while RSA is an asymmetric cryptographic algorithm. Encryption and decryption is done with a single key in AES whereas uses separate keys (public and private) in RSA. AES uses 10, 12 or 14 rounds depending on the key size 128, 192 or 256 respectively.

ISSN: 2321-9653

4. RESULT ANALYSIS

4.1 Level of security

With a key of length n bits, there are 2^n possible keys. Even if a cipher is unbreakable by exploiting structural weaknesses in the algorithm, it is possible to run through the entire space of keys which is known as brute force attack. AES-256 is more secure than RSA-128 because it has 256-bit key - that means 2^{256} possible keys to brute force, as opposed to 2^{128} in RSA-128. RSA-128 is vulnerable and can be easily broken by brute force attack in merely 15 hours. Even though RSA used separate keys for encryption and decryption it has only one round in the algorithm. But AES uses 10, 12 or 14 rounds which further improve the security and reduce brute force attack.

Also it is not good to increase the size of key in RSA encryption. If you want to use 1024 bit RSA then the modulus integer n will have 1024 bits. This means two prime numbers of roughly 512 bits each must be generated. Doubling the size of key will increase the time required for encryption operation by a factor of 4 and decryption by a factor of 8 because d changes on direct proportion to the size of modulus.

AES-encrypted data is unbreakable in the sense that no known cryptanalysis attack can decrypt the AES cipher text without using a brute-force search through all possible 256-bit keys. But breaking a symmetric 256-bit key by brute force requires 2^{128} times more computational power than a 128-bit key. 50 supercomputers that could check a billion billion (10^{18}) AES keys per second (if such a device could ever be made) would, in theory, require about 3×10^{51} years to exhaust the 256-bit key.

4.2 Encryption and Decryption time

The encryption time is considered the time that an encryption algorithm takes to produces a cipher text from a plain text. Encryption time is used to calculate the throughput of an encryption scheme, is calculated as the total plaintext in bytes encrypted divided by the encryption time. The time taken by RSA for encryption and decryption in is much higher than the time taken by AES.

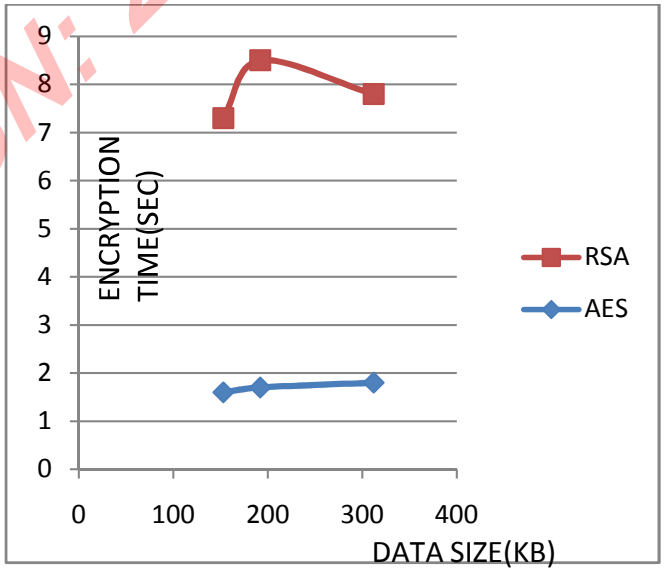

Fig 4.1: Comparison of encryption time

From fig 4.1 it is clear that the encryption time required for a given data is high for RSA than AES. Thus along with brute force and mathematical attacks, RSA algorithm also faces high computation burden. It is because in RSA the modulus n must be selected in such a manner that the following is guaranteed.

$M^{ed} = M \pmod n$

We want this guarantee because $C = M^e \bmod n$ is the encrypted form of message and decryption is carried out by $C^d \bmod n$. It is guaranteed when n is the product of two prime numbers n=p*q where p and q are individually as well as relatively prime. If so we can decompose the function of n into product of functions of p and q.

$f(n) = f(p) * f(q) = (p-1) * (q-1)$

When small values of p and q are selected for designing the key, the encryption process become too weak. If p and q lengths are large then more time is consumed for encryption and decryption. Thus it introduces high computation overhead in RSA algorithm.

5. CONCLUSION

Seccloud protocol is the best known protocol to ensure security and privacy for storage and computation in cloud computing. Data is first encrypted at the cloud user and then send to the cloud. This data is decrypted and stored at the cloud server. For this RSA algorithm was used initially. It is an asymmetric algorithm that use public and private key. RSA encryption is best suited as a key transport algorithm. But it suffered from brute force attack and other mathematical errors. RSA also consumed high encryption and decryption time while execution. This becomes serious when the amount of data to be encrypted is large. In order to overcome this AES algorithm was introduced which consumed less encryption and decryption time and also is not vulnerable to brute force attacks. AES is a symmetric block cipher, and is incredibly fast. The algorithm is based on several substitutions, permutations and linear transformations each performed on a block therefore called as block cipher. These operations are repeated for several stages called rounds. A change in a single bit in key or plain text results in a completely different cipher. This is an advantage over traditional stream ciphers. Also AES-256 is more secure than RSA-128 because it has 256-bit key - that means 2^{256} possible keys to brute force, as opposed to 2^{128} in RSA. Thus AES remains as the preferred algorithm for implementing Seccloud protocol to ensure security and privacy in cloud computing.

6. REFERENCES

A. Juels, B. Kaliski Jr., “PORs: proofs of retrievability for large files” (2007)

B. Kang, C. Boyd, E. Dawson, “A novel identity-based strong designated verifier signature scheme” (2009)

C. Wang, K. Ren, J. Wang, “Secure and practical outsourcing of linear programming in cloud computing” (2011)

Cong Wang, Kui Ren, Wenjing Lou, Jin Li, “Toward Publicly Auditable Secure Cloud Data Storage Services” (2010)

Cong Wang, Qian Wang, Kui Ren, Ning Cao, Wenjing Lou, “Towards Secure and Dependable Storage Services in Cloud Computing” (2009)

G. Ateniese, R. Di Pietro, L. Mancini, G. Tsudik, “Scalable and efficient provable data possession” (2008)

Jachak K.B, Korde S.K, Ghorpade P.P, Gagare G.J “Homomorphic authentication with random masking technique ensuring”(2012)

Jiawei Yuan, Shucheng Yu, “Secure and Constant Cost Public Cloud Storage Auditing with Deduplication” (2012)

K.Govinda, V.Gurunathaprasad , H.Sathishkumar, "third party auditing for secure data storage in cloud through digital signature using rsa" (2012)

K.RaviTeja, Srinivasa Narasanna Pilli, B.Sreenivasa Rao, M.JangaReddy, "Secure Storage in Cloud Computing & Emergence of Intruder Detection" (2012)

L. Wei, H. Zhu, Z. Cao, W. Jia, A. Vasilakos, "Seccloud: bridging secure storage and computation in cloud" (2010)

M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, "A view of cloud computing" (2010)

M. Belenkiy, M. Chase, C. Erway, J. Jannotti, A. Küpçü, A. Lysyanskaya, "Incentivizing outsourced computation" (2008)

M. Castro, B. Liskov, "Practical byzantine fault tolerance and proactive recovery" (2002)

M. Jakobsson, K. Sako, R. Impagliazzo, "Designated verifier proofs and their applications" (1996)

Ning Cao, Shucheng Yu, Zhenyu Yang, Wenjing Lou, Y. Thomas Hou "LT Codes-based Secure and Reliable Cloud Storage Service" (2012)

P. Golle, I. Mironov, "Uncheatable distributed computations" (2001)

Prerna, Abhishek, "A study of encryption algorithms AES, DES, RSA for security", (2013)

Q. Wang, C. Wang, J. Li, K. Ren, W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing" (2009)

Qin Liua, Guojun Wanga, Jie Wub, "Secure and privacy preserving keyword searching for cloud storage services" (2011)

R. Canetti, B. Riva, G. Rothblum, "Verifiable computation with two or more clouds" (2011)

Reza Curtmola, Osama Khan, Randal Burns, "Robust Remote Data Checking" (2008)

S. Pearson, Y. Shen, M. Mowbray, "A privacy manager for cloud computing" (2009)

Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing" (2010)

W. Du, J. Jia, M. Mangal, M. Murugesan, "Uncheatable grid computing" (2004)

Yan Zhu, Huaixi Wang, Zexing Hu, Gail-Joon Ahn, Hongxin Hu, Stephen S. Yau, "Efficient Provable Data Possession for Hybrid Clouds" (2010)

Zhuo Hao, Sheng Zhong, Nenghai Yu, "A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability" (2011)