A Study on Key Integrity and Data Security Verification for Cloud Computing Environment

Pruthviraj R. Pawar[1]* Dr. Santosh Kumar Mishra[2]

[1] Research Scholar, Computer Science Engineering, Maharishi University of Information Technology University, Lucknow

[2] Assistant Professor, Computer Science Engineering, Maharishi University of Information Technology University, Lucknow

Abstract – Cloud computing is one of the most powerful innovation that has caught fancy of technologists around the globe. Cloud Computing works as "Pay-as-you-go" model is most appealing factor. Cloud Computing provides the capability to use computing and storage resources on a metered basis and reduce the investments in an organization's computing infrastructure. Cloud Computing has enormous advantages such as scalability, rapid elasticity, measured services, & most important of them the potential that it has for cost savings to the enterprises, it also has its share of security risks that no enterprise can afford to overlook. Security and risk assessment would encompass analysis of the impact of variety of threats and attacks on various aspects of cloud computing including; Adaptation of Cloud computing, maintenance of secrecy and privacy of personal data, access and updating of data.

Keywords: Key Integrity, Data Security, Cloud Computing.

- X -

INTRODUCTION

Cloud computing is defined as the utilization of the computer resources in an on-demand basis over the network. The cloud provides the dynamic computing environment for the end users, and also provides a global network platform for executing the processes virtually across the network. The services such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) can be offered in the virtualized platforms. The data that is provided by the Cloud Service Providers (CSP) are accessed in the form of multiple virtual servers. The movement of data to the centralized servers for storing the files in the cloud could affect the user's interaction privacy, and security. Thus, the data integrity is very important for verifying the validity of the data. The hosting of an application in the cloud environment introduces two main security issues including data security, & code security. The cloud computing environment reduces the cost, and also provides high service quality. The important qualities for the cloud users are depicted in the figure 1.

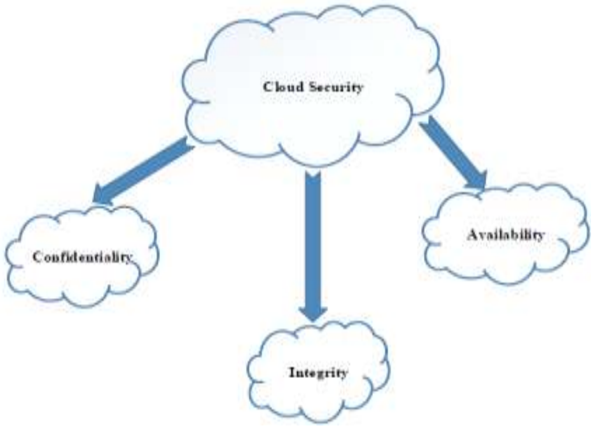

Figure 1. Issues in the Cloud Computing Environment

Confidentiality

Data confidentiality is the process of preventing the user data from unauthorized access. The confidentiality can be ensured using the security protocols, services of authentication, and data encryption.

Integrity

The data integrity ensures that the data transmitted, and the data received are same. At the time of transmission, the data cannot be reformed in-between. If the data that is transmitted is not similar to the data that is received, then the integrity is considered to be lost. The integrity of the data is assured using the firewalls, and intrusion detection.

Availability

The availability of the data ensures that the data is accessible to the user without any interruption. Further, irrespective of the user location, the availability of the data can be ensured using fault tolerance, authentication, and network security. Generally, to protect the data in the cloud, the cryptography techniques are used. As shown in the figure 2. The cryptography technique includes two main processes such as encryption, and decryption. The encryption process is mostly preferred by the sender. The encryption process converts the clear text of the sender into cipher text using the encryption algorithms. On the other hand, the decryption process converts the cipher text into plain text using the decryption algorithms.

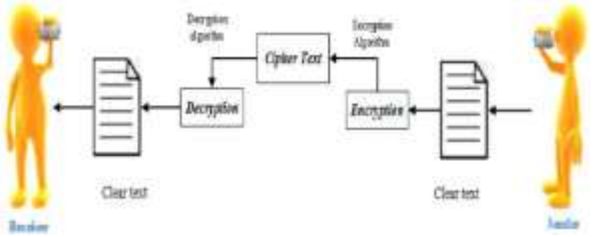

Figure 2. Example of the encryption and decryption process

When the cryptographic techniques are used for the data transmission, the eavesdropper cannot crack the information that is sent by the sender. Some of the security issues that are involved in the cloud computing model are depicted in the figure 3.

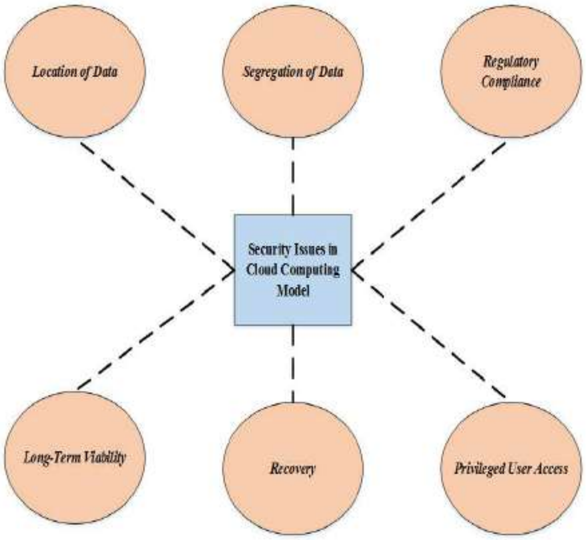

Figure 3. Security Issues in Cloud

The figure 4 depicts the key security attention areas of the cloud. To provide security for the data at rest, the cryptographic encryption mechanisms are the optimal option. Securing the data using software encryption is very slow, and less secure. Hence, the hard drive manufacturers purchase the self-encrypting drives that deploy the trusted storage standards of the trusted computing group. The self-encrypting drive builds the encryption hardware into the drive, and provides automated encryption with less cost, and high performance. To protect the data at transmission, the encryption is considered as the optimal option. Further, the authentication, and integrity protection mechanisms are used to make sure that the data is prevented from modifications. The legal, and the regulatory issues of the cloud computing creates multiple impacts. To validate that the cloud provider has strong policies, and practices for addressing the legal and regulatory issues, the customers get the guidance of legal, and regulatory experts by inspecting the cloud provider's policies, and practices.

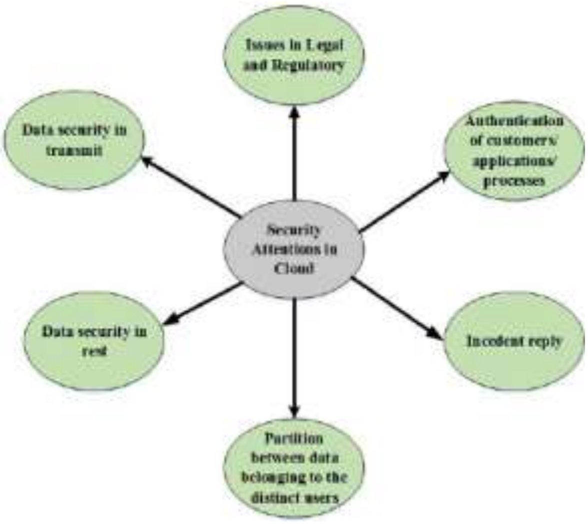
Issues in Legal and Regulatory
Data security in transmit
Authentication of customers/ applications/ processes
Security Attentions in Cloud
Data security in rest
Incodent reply
Partition between data belonging to the distinct users

Figure 4. Security attention areas of cloud

The experts consider the data security, and export, compliance, auditing, data retention and destruction, and legal recovery. The existing data security approaches the secured data using cryptographic solutions, and random key generation processes. But, the data integrity, and robust key generation processes are not there. Further, the computational overhead that occurs during the encryption and decryption process is high, because the cloud handles the huge amount of data at the same time. Handling data at the same time by various users results in key collision. The key loss thus crashes the original data that is provided by the data owner in the cloud server. In this research, to address all these issues, a data mechanism that focuses on the data security is proposed. Initially, a Key Derivation Policy (KDP) is proposed for enhancing the data security, the Advanced Encryption Standard (AES) is used for the encryption, and the decryption processes. The integration of the KDP with the AES enhances the security. The integrity of the key is validated using the Hash Message Authentication Code (HMAC). Further, the key loss is managed using the HMAC.

The working principle of the proposed data security, and key integrity verification using key derivation policy is illustrated in the following sections.

DATA SECURITY AND KEY INTEGRITY VERIFICATION USING KEY DERIVATION POLICIES

The overall flow of the proposed Advanced Encryption Standard (AES) based Key Derivation Policy (KDP) is depicted in the figure 5.

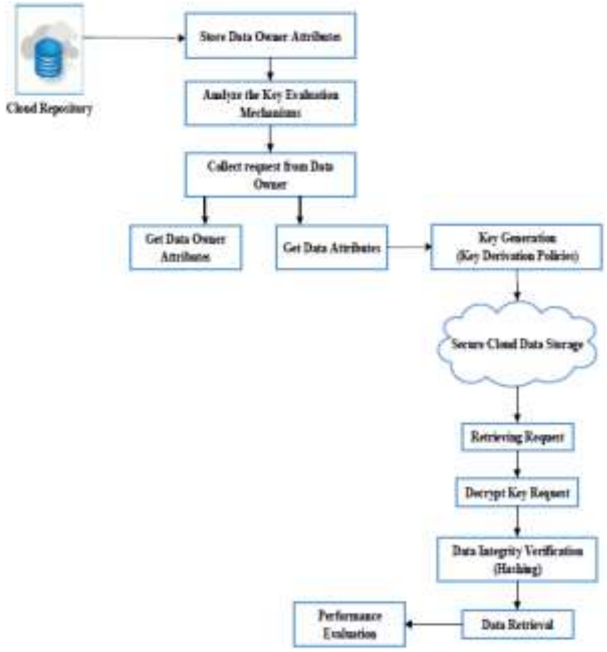
Cloud Repository
Store Data Owner Attributes
Analyze the Key Evaluation Mechanisms
Collect request from Data Owner
Get Data Owner Attributes
Get Data Attributes
Key Generation (Key Derivation Policies)
Secure Cloud Data Storage
Retrieving Request
Decrypt Key Request
Data Integrity Verification (Hashing)
Performance Evaluation
Data Retrieval

Figure 5. Overall Flow of the Proposed Method

The proposed method includes has three important components such as

- Analysis of Key Evaluation Mechanism
- Key Derivation Policies (KDP)
- Verification of Key Integrity (HMAC)

The detailed description about each component is illustrated in the following sections.

Analysis of Key Evaluation Mechanism

The Attribute Based Encryption (ABE) is used for one-to-many encryption. By exploiting the attributes, the public key for encrypting the data is generated. In ABE, there exist three main factors such as authority, data owner, and data user. The role of each actor is illustrated in the figure 6. The figure shows that the main role of the authority is to generate keys for the data owners and the data users for encrypting or decrypting the data. The authority generates the public key, and master key based on the attributes. The generated keys are maintained for the future access. If a new data user enters the system without the pre-defined attributes, the authority will redefine the attributes and will regenerate the public key, and master key. The main role of the data owner is to encrypt the data with the public key, and the set of attributes.

The data owner maintains the data for sharing. The data owner initially registers their data in the cloud server, and obtains their access by authorizing the credential information from the cloud server. The data owners can process or create the data file. Further, they can generate the data repository for deriving the data owner attributes. The role of the data user is to decrypt the encrypted data using the private key received from the authority. While decrypting the data, the attributes in the private key of the data user, and the attributes in the encrypted data should be matched. If there exists a match, the data is decrypted else the data user is not allowed to decrypt the data. In this phase of the research, the ABE exploits the attributes of the data, and the data owner for encrypting, and decrypting the messages. The information related to the data owner are preserved, managed, and backed up in the cloud repository. The data owners access the cloud repository through the internet. Subsequently, the key evaluation mechanism is analyzed for the security purpose. After analyzing the key evaluation mechanism, the attributes of the data, and the data owner are obtained.

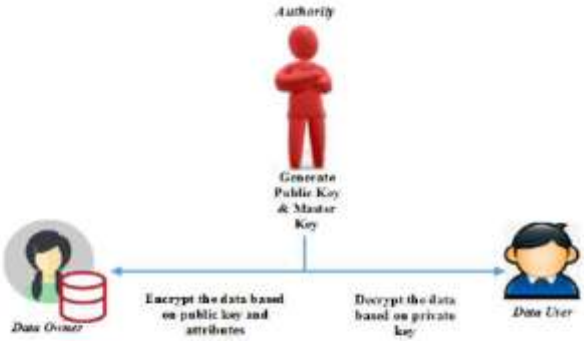
Authority
Generate Public Key & Master Key
Data Owner
Encrypt the data based on public key and attributes
Decrypt the data based on private key
Data User

Figure 6 Attribute Based Encryption Mechanism

Key Derivation Policy (KDP)

To address the security issues of the cloud such as complexity of the system, shared multi-tenant environment, and control loss, the cryptographic technique is exploited. The security model of the proposed system includes three processes such as,

- Encryption

- Key generation
- Decryption

Encryption

The encryption is the process of converting the plain text into the cipher text that could not be understood by anyone except the authorized person. The data is often encrypted using the encryption algorithm, and an encryption key. The encryption process creates the cipher text that could be viewed in the original form if decrypted using the correct key. The encryption process can be executed by anyone who wants to encrypt the data. As shown in the figure 7, the encryption process includes the following parameters for encrypting the data.

- Public-key
- An index that denotes the class of the cipher text
- Input message

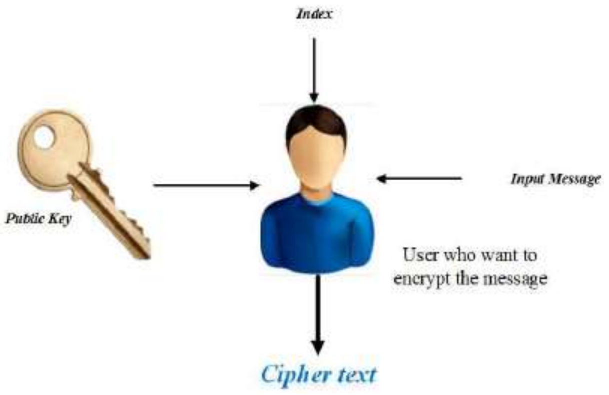

Figure 7. Encryption of the Input Message

Decryption

The decryption is the process of converting the cipher text into the plain text. The data is often decrypted using the decryption algorithm, and the decryption key. The decryption process is executed by the delegate who received the aggregate key that was generated by the extract process. In order to decrypt the cipher text on the receiver end, the parameters in the figure 8, are exploited.

- Aggregate key
- An index that represents the class of the cipher text
- Cipher text

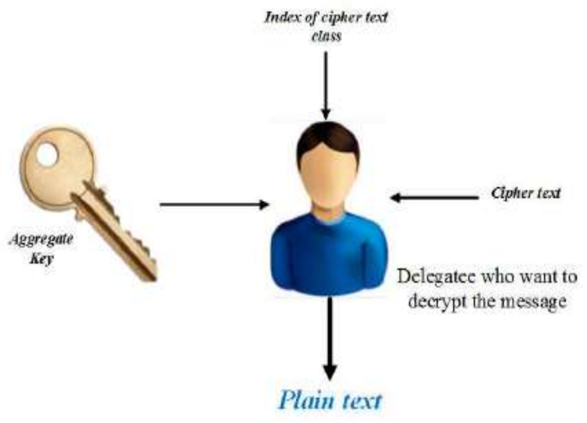
Index of cipher text class
Aggregate Key
Cipher text
Delegatee who want to decrypt the message
Plain text

Figure 8. Decryption of the Cipher Text

The typical structure of the AES contains three fixed 128-bit block ciphers with the cryptographic key sizes such as 128, 192, and 256 bits. The AES algorithm begins with add round key stage, later it performs 9 rounds of four stages, and tenth round of three stages. The four stages that are involved in the encryption, and decryption processes are,

- Substitute bytes
- Shift rows
- Mix columns
- Add round key

The tenth round leaves out the mix column stage. The first nine rounds of the decryption process are,

- Inverse shift rows
- Inverse substitute bytes
- Inverse Add Round Key
- Inverse Mix Columns

The steps involved in the encryption, and decryption processes are same. The decryption process performs the inverse operation of the encryption.

Substitute bytes

The substitute byte is a lookup table that contains a 16x16 matrix byte values. The matrix is composed of all the possible combinations of the 8-bit sequence. The matrix that is used for the entire encryption process is called as the state. At each round, the leftmost nibble of the byte specifies a particular row of the s-box, and the rightmost nibble of the byte specifies the column. The inverse substitute byte transformation exploits the inverse s-box for processing.

Shift Row Transformation

This stage computes only the simple permutation. The shift row transformation maintains the first row of the state unaltered. The second row is shifted one byte left in the circular manner, the third row is shifted 2 bytes to the left in the circular manner, and the fourth row is shifted three bytes to the left in the circular manner. The transformation of the inverse row of shifts takes place for each of the last three rows in the opposite direction.

Mix Column Transformation

The mix column stage is a substitution stage. Each column is processed individually. The byte in the column is mapped into a new value. The elements in the product matrix are obtained by adding the sum of the products of elements of 1 row, & 1 column,

Add Round Key transformation

In this stage, the 128 bits of the state are bitwise XORed with the 128 bits of round key. A column wise operation is performed between 4 bytes of the state column, & one word of the round key. This Add Round key transformation improves the efficiency of every bit of the state.

In this research, to enhance the security, the similar key, or the master key is used for computing the cipher at the secure end level. The AES encryption/decryption can be performed by using either the symmetric key or the asymmetric key. The suggested cryptographic technique used the symmetric key based AES algorithm because it consumes less power for computation, and also minimizes the burden for the key management. Generally, the private key is maintained in secret by the data owner but the public key is distributed to others for encryption. By exploiting the encryption algorithm, the data of the user can be made secure in the cloud environment. As the cryptographic techniques such as the proprietary algorithms do not provide sufficient security, it is not trusted much. The symmetric AES technique provides stronger encryption with the longer key length. These keys are computationally intensive and also provide better protection. The cloud users submit their service request to the Cloud Service Provider (CSP). Among the available CSPs, the optimal service provider is chosen by the cloud user. Once an application sends the data to the cloud, the symmetric AES algorithm encrypts it, and uploads it back to the cloud. This type of encryption protects the data, and the encryption keys. The steps involved in the proposed AES encryption, and decryption algorithm is illustrated below,

Secure Cloud Data Storage

The important concern in the cloud computing environment is to protect the files, and data from the unauthorized user. Storing the data in third party cloud system may affect the data confidentiality. To address this issue, the data is encrypted, and stored in the storage server. The data owner encrypts the data using the cryptographic methodology, and stores the encrypted data on the cloud storage server. Though the encryption of the data provides data confidentiality, it does not provide high security, and dynamic data modification. The malicious user can hack the data while transferring the data from the data owner to the cloud server or the attacker can decrypt the data directly from the cloud server using the cryptographic keys.

On obtaining the data, the attacker can perform certain modifications, and store back the modified data in the storage server like the data owner. As the data looks same as the original data, the cloud users, and the data owners cannot find whether the data is hacked. Hence, to address all these issues, the data owner generates a private/master key pair through the key generation. The encryption process encrypts the messages, and also chooses which cipher text class should be connected with the plain text message. By using the master-secret keys the aggregate decryption key is generated. The created keys are passed to the delegates through the secure devices. At last, the user with the aggregate key decrypts the cipher text. The figure 9 represents the process of secure data storage in the cloud environment. The steps involved in the secured cloud storage are depicted in the figure 9.

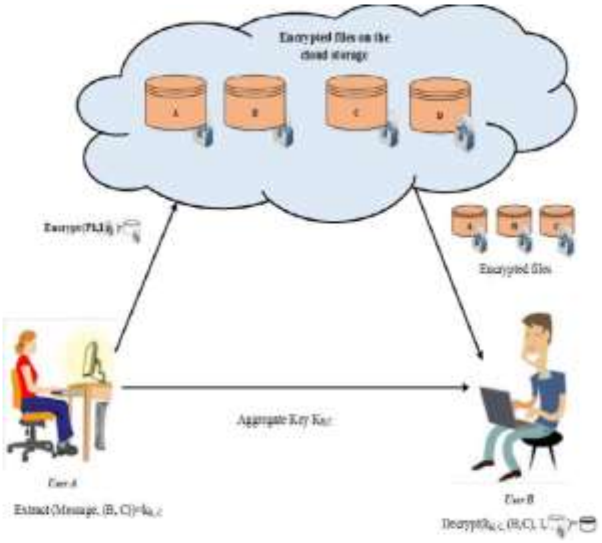

Figure 9. Secure Data Storage in Cloud

Attacks in the Hash Based Message Authentication Protocol

Based on the birthday paradox, the attacks in the HMAC are classified as follows,

- Collection of $2^{1/2}$ random messages with a b-bit length that is denoted as N_j, and the ask values for the MAC values that is denoted as B_j.

- Computation of message pairs N_j and N_i such that $B_j = B_i$

For each of the (N_j , N_i) pairs, the MAC pair such as (N_j ‖ P), and (N_i ‖ P is computed. The P denotes the non-empty string. If any of the MAC pair collides then the output of the MAC algorithm will be equivalent to the HMAC.

The key attacks that occur in the HMAC are depicted in the figure 10.

Distinguishing attack

The figure 10 shows the classification of the distinguishing attacks. The distinguishing-R attack distinguishes the HMAC from the random function. When the cryptanalyst want to validate that the output, strings are produced from the HMAC, thus distinguishing-R attacks which will be beneficial.

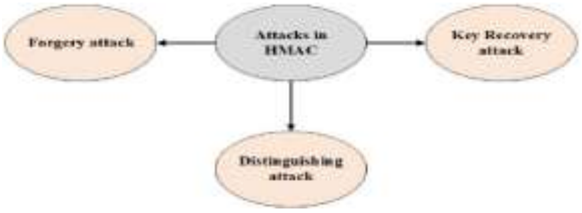

Figure 10. Attacks in the HMAC

This attack consumes $2^{1/2}$ number of messages, where ` denotes the length of initial value. It also consumes a probability of 0.63. The distinguishing H-attack distinguishes the HMAC constructed by the hash function from the HMAC generated by the random function. When the cryptanalyst wants to validate that the cryptographic Hash functions embedded with the HMAC, the distinguishing-H attacks will be useful. Among the two types of distinguishing attacks, the distinguishing R-attack can be instantly converted into the forgery attack. In certain cases, the distinguishing attacks detect the innermost collision with differential paths of the cryptography that is embedded with the HMAC.

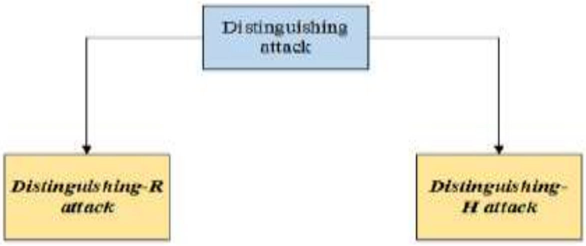

Figure 11. Types of Distinguishing attack

Based on the structure of the HMAC, the distinguishing H-attack can be classified into differential distinguisher, and rectangle distinguisher. The complexity of the distinguishing attack is O(2^{w+1})queries.

Forgery attack

The forgery attack is the most common attack in HMAC. In case of the iterative hash function, the distinguishing attack introduces a forgery attack with one additional chosen query. By exploiting this attack, the opponent produces an effective message/tag pair without having much knowledge about the secret key.

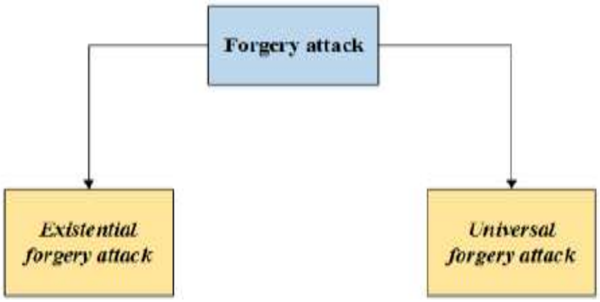

Figure 12. Classification of Forgery attacks

The figure 12 shows the classification of the forgery attack. The existential forgery attacks compute the valid MAC for the random message. It provides the ability to generate the message, and signature pair. The messages that were not signed in the past by the legitimate owners are chosen. Most of the existing digital signature algorithms allow the existential forgery. The universal forgery attack computes the valid MAC for any given message. Among the two types of forgery, the universal forgery is the malicious one.

Key recovery attack

If the Hash function has the same step function as the MDs, then the collision path recovers the inner key in the HMAC. The query complexity of the key recovery attack is O(2^{w+1}). The time complexity of the key recovery attack depends on the collision path form. In the key recovery attacks, each bit of the collision information represents one bit of the inner key. The collision information contains the entire information about the hash computation. The analyses of these threats require a huge number of message pairs. Once the key integrity verification is completed, the data is retrieved and the performance is analyzed.

- To address all these issues, the Key Derivation Policy (KDP) along with the Adaptive Encryption Standard (AES) is proposed.

- The suggested methodology focuses on the security of the key for the data that is been outsourced in the cloud servers.

- Further, it provides secured access control mechanism, and data access policies for improving the data security in cloud.

- The key loss is prevented using the key integration verification technique named Hash Message Authentication Code (HMAC).

- The usage of the HMAC enhances the data security in cloud computing environment.

- To prove the effectiveness of the proposed KDP, it is compared with the existing subset cover for the security performance metric.

- The encryption time of the proposed DKP is compared with the existing role based access control.

- The variation of the encryption time with respect to the file size is analyzed. The time consumption for the proposed key generation process is compared with the existing methodology such as fully Homomorphic key generation, and Fast Fourier Transform (FFT).

- The analysis of the decryption time with respect to the number of attributes is analyzed for the existing Attribute Based Encryption (ABE), and the proposed DKP methods.

- Further, the encryption/ decryption time, and encryption speed for the existing KP-ABE, CP-ABE, CP-ABE-WP methods are compared with the proposed DKP.

- The performance analysis results show that the proposed DKP provides less encryption time in regard of file size.

- Further, it provides optimal encryption speed than the existing methods. When the number of requests rises, the KDP will fail by restriction of the attributes.

- Hence, in the next chapter, the proposed KDP integrates the attributes in the attribute-based encryption with the several numbers of requests.

CONCLUSION

Cloud computing has introduced multiple services, and also has reduced the IT cost. Protecting the data that is stored in the cloud computing environment is vital. In case of the traditional data protection approaches, the network-centric, and perimeter security are often used with the devices such as firewalls, and intrusion detection systems. But the existing data protection solutions lack the data integrity, robust key generation process, computational overhead, collision in the encryption key, and data crashing by the data owner in the cloud server. We conclude once the key integrity verification is completed, the data is retrieved and the performance is analyzed.

REFERENCES

[1]. Tirthani N & Ganesan R (2014). 'Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography', IACR Cryptology ePrint Archive, Vol. 2014, P. 49.

[2]. Tirthani, N & Ganesan, R (2014). "Data security in cloud architecture based on diffie hellman and elliptical curve cryptography", IACR Cryptology e-Print Archive, vol. 2014, pp. 1-5.

[3]. Singla S & Singh J. (2013). 'Survey on Enhancing Cloud Data Security using EAP with Rijndael Encryption Algorithm', Global Journal of Computer Science and Technology, vol. 13, no. 5, pp. 11-14.

[4]. Sood, SK (2012). A combined approach to ensure data security in cloud computing", Journal of Network and Computer Applications, vol. 35, no. 6, pp. 1831-1838.

[5]. Selvamani, K & Jayanthi, S. (2015). A review on cloud data security and its mitigation techniques", Procedia Computer Science, vol. 48, pp. 347-352.

[6]. Puneetha, D. (2014). Data security in cloud using elliptic curve cryptography", International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, no. 5, pp. 4187-4192.

[7]. Kaur, M & Mahajan, M. (2013). Using encryption algorithms to enhance the data security in cloud computing", International Journal of Communication and Computer Technologies, vol. 1, no. 12, pp. 56-59.

[8]. Kant, DC & Sharma, Y (2013). Enhanced security architecture for cloud data security", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 5, pp. 571-575.

[9]. Gampala, V, Inuganti, S & Muppidi, S. (2012). Data security in cloud computing with elliptic curve cryptography", International Journal of Soft Computing and Engineering (IJSCE), vol. 2, no. 3, pp. 138-141.

[10]. Chen, D & Zhao, H. (2012). Data security and privacy protection issues in cloud computing", International Conference on Computer Science and Electronics Engineering (ICCSEE), pp. 647-651.

Corresponding Author

Pruthviraj R. Pawar*

Research Scholar, Computer Science Engineering, Maharishi University of Information Technology University, Lucknow