Volume: 6 Issue: IV Month of publication: April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4203

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

Securing Cloud Data with Pseudonym based Cryptography and Data Deduplication for Cloud Storage

Lakshmi Sravanthi Guthikonda[1], Asst Prof K. Naresh[2]
[1, 2] *School of Computer Science and Engineering (SCOPE), VIT University*

Abstract: To provide security of data, the cloud storage service need to implement effective access control mechanism to users. Attribute based encryption (ABE) is an optimistic cryptographic access control technique to ensure the end-to-end security of data in cloud environment. The existed attribute based encryption researches mainly focusing on the efficiency decryption. While creating anonymous communication system and key distribution for users. Users required to register at organizer to drive their private keys. For sharing public key central authority using communication medium like mail. There may be a chance to hack that keys. To avoid this problem, we present a new label based access control model by using pseudonym key, with multi authorities to describe detailed relationship of entities and DNA is cryptography system and data deduplication in our scheme. Data duplication compares objects and removes objects that already exist in the data set.

Keywords: DNA, pseudonym key, duplication

I. INTRODUCTION

Cloud computing provides virtually accessible infrastructure to users on that cloud environment users can store their files and execute applications. But cloud computing has security challenges. The main reason is cloud operators store and handle user data outside of the reach of users. To provide security there are various approaches to extend cryptography to cloud environment and data can only be decrypted by authorized parties that have access to the appropriate decryption keys. Still facing security issue while sharing the keys. For example, while sharing applications in organization and sharing generated keys based on attributes by using communication medium. Might be a chance to stolen by unauthorized. Users or some malwares. To avoid this problem we are using pseudonym cryptography. In this which key manually given by data owner and submit them to the organizer and then the organizer generates corresponding private keys and sends them back to the user. This is because that a pseudonym can simultaneously serve.

A. For two Purposes

1) A public key and
2) Hash key (identity)

Data deduplication is a technique to identify the file or data of same contents and only store one copy of them. Therefore, data deduplication can cost of the cloud storage capacity and utilize cloud storage more properly. According to the original cloud storage schemes, one of schemes is to store the whole file into the storage server without any deduplication. Thus, if here are two similar files, the cloud storage server would store redundant blocks between these two similar files. Therefore, the cloud storage capacity cannot be used properly. To avoid that problem cloud storage vendors using the technique of data deduplication when storing the uploaded files, the Drop Box for example. Some data deduplication schemes calculate a hash value for each file used to check whether there is redundant hash value among uploaded files in the cloud storage. Others translate a file into n blocks and then calculate a hash value to represent every block; therefore, the cloud storage server can examine the redundancy of every hash value of blocks. Though this method can find those blocks not stored in the cloud storage server, it spends too much time on examining these duplicate blocks.

Attribute-Based Encryption (ABE) has emerged as a promising technique to ensure the end-to-end data security in cloud storage system. It has significant advantage over the traditional PKC primitives and data properties as it achieves flexible one-to-many encryption instead of one-to-one. In ABE, the cipher texts and encryption keys are characterized with sets of descriptive attributes defined for the system users. A sender first defines access policies for its data, and then encrypt the data file to others who have a certain set of attributes drawn from a pre-defined attribute universe, then a receiver could use its own private key to decrypt this ciphertext only if its attributes satisfy the pre-defined policy. ABE thus is envisioned as an important tool for addressing the problem of secure and fine-gained data sharing and access control, so that it has attracted much attention in research community.

In order to protect information, secret writing was used since ancient times and is used as well nowadays. Well known and widely used techniques which implement secret writing are cryptography and steganography. These two sciences manipulate information in order to cipher or hide its existence. DNA has a great cryptographic strength, its binding properties between nucleotides bases (A—T, C—G) offer the possibility to create self-assembly structures which are an efficient means of executing parallel molecular computations; its storing capabilities are enormous, a gram of DNA includes 1021 bases equivalent to 108 terra-bytes. Actual implementations don’t exceed laboratory level, are expensive and require time. Simple and effective algorithms are quested in order to bring DNA computing on digital level and use it on large scale. This paper presents two original DNA cryptographic algorithms based on existing ideas described in related literature: public key encryption of binary data followed by its transform in DNA digital sequence;

II. RELATED WORK

The Attributes based encryption researches mainly focusing on the efficiency decryption means flexibility of policy, the cost of communication, and the cipher texts of metadata management are still challenging issues. In a hybrid Cloud Environment like organization details, the Centralized system might use by different employees say A and B working on running their applications. Here the organization’s personal are sensitive data may be prone to risk. In those cases, user secret keys cloud is easily hacked or used by an unauthorized party or else any malwares. Even though the computer may be secured by a password. In such cloud environments a more secure way is to use two factor authentications. Two-factor authentication is very common among web-based e-banking services. In addition to a username/password, the user is also required to have a device to display a one-time password like mobile. We proposed a new distributed, scalable and fine-grained access control scheme based on pseudonym cryptography. The classification attributes and threshold policies are integrated into an access structure, and then the objects are encrypted with the integrated access structure. The constant-size cipher text components related to attributes can be managed as the corresponding metadata. As a result, the encryption complexity and cipher text storage are reduced.

III. SYSTEM MODEL

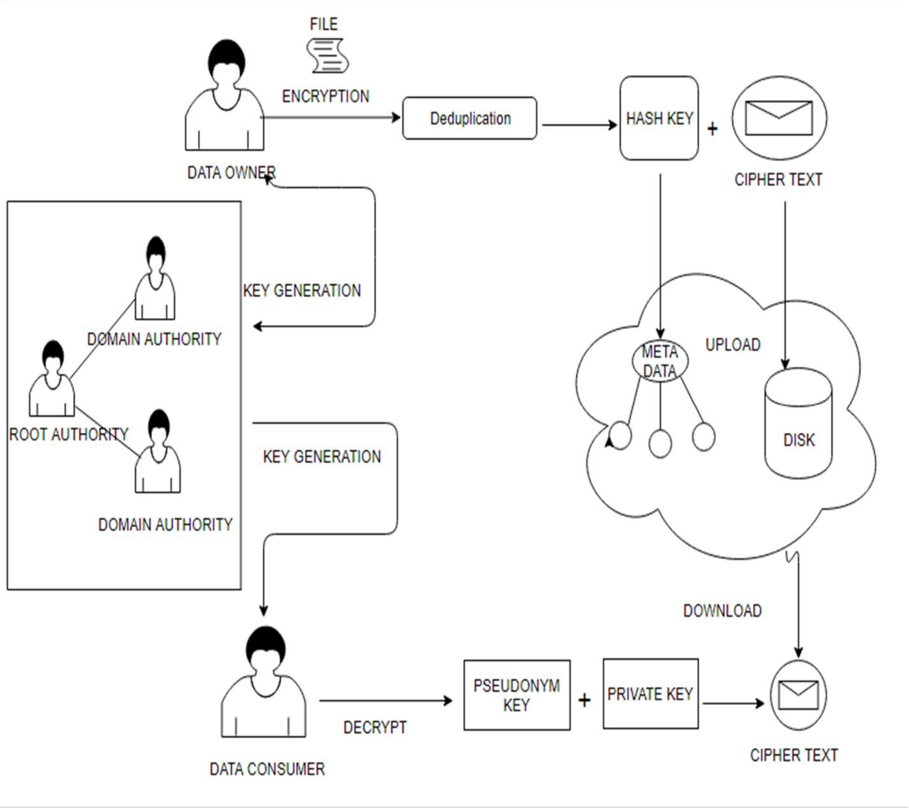


Fig1: System Model

A. System Entities

The system model in cloud is given in Fig.1, which consists of five parts: authorities, a cloud service provider (CSP), data owner, data consumer and object.

B. Authorities

authorities are in charge of distributing attribute private keys, which consist of two hierarchical components, as shown in Fig.1. The root authority is responsible for system initialization and domain authority's management. While, the domain authority is managed by its parent domain authority or the root authority. Each higher domain authority corresponds to a higher source classification attribute sets.

C. Cloud Service Provider (CSP)

It provides the object-based storage system service, include the metadata management.

D. Data Owner

In the system, the owner outsources its data in the cloud by the write rule for sharing with others. Meanwhile, it is in charge of defining the metadata to the CSP.

E. Data Consumer

The data consumer takes the read rule to access its interest data in the cloud.

1) Object: It is the data that the owner wants to protect. And the attributes is also defined in its label by the owner.

Model The system model (Fig.1) consists of a root authority, multiple domain authorities, and numerous data owners and data consumers. Root authority is responsible for generating the system parameters and private key and also pseudonym key to authorize the top level domain authorities, and each domain authority uses DNA key gen and deduplication algorithms to delegate keys for the subordinate domain authorities or data consumers. Before any process operations, each participant should register into the system. During the registration, every consumer receives its secret keys corresponding to the attributes from domain authority by using DNA Key Gen algorithm. and pseudonym key (hash code) corresponding to the file from domain authority by using MD5 algorithm.

Then, the data owner processes the object as follows: Firstly, the data owner chooses a symmetric content key to encrypt object The ciphertext is denoted as M. After that, the data owner encrypts data with access policy by Encrypt algorithm. Finally, the owner uploads the integrated and object label as metadata. The procedures of read is described as below. First, the data consumer downloads object's metadata, and decrypts. Then it obtains the pseudonym key by using md5 and private key by using DNA Decrypt with its subject label and the object label from metadata. Finally, the consumer can access the data M by using symmetric decryption algorithm.

There are some problem assumptions should be considered:

2) CSP and authorities are always online.

3) The CSP is curious and cannot be trusted completely. It may collude with malicious users to harvest contents of big data for its own benefit.

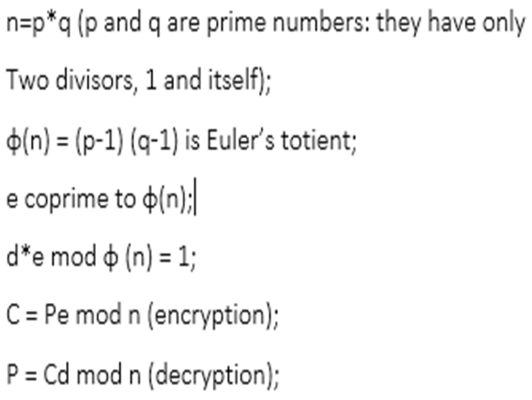
4) Data consumers can access object for reading only.

IV. PROPOSED CONSTRUCTION

Deduplication aims to seek out as several redundancies attainable whereas maintaining interval. to cut back interval, one typical technique is to see indexes of knowledge in memory before accessing disks. If the information indexes area unit a similar, deduplication doesn't involve accessing the disks wherever the indexes area unit hold on, which might cut back interval we have a tendency to show associate degree implementation of associate degree index computation exploitation associate degree md5 hash perform therefore, we have a tendency to show codes to reckon a md5 hash key from a file and knowledge. We have a tendency to develop a wrapper category with 2functions, like get Hash Key of File and acquire Hash Key (string data). The generated hash key owner can serve to knowledge client as an anonym key. Client can get non-public key from knowledge owner by exploitation deoxyribo nucleic acid key generation in any communication medium that registered by organization. Knowledge client will ready to decode by these 2 factors, couldn't with one issue.

A. Key Generation

The plaintext message is encrypted with RSA public key algorithm. The security of this algorithm is given by the computational difficulty of factoring large numbers. To be secure, very large numbers must be used as primes, 100 decimal digits at the very least. Product of such large prime numbers is an easy mathematical operation, but reverse process is a very hard task. It is extremely difficult, nearly impossible, to determine the original values from the product, at least it will take a lot of time. This algorithm offers the public key (n, e) for encryption and the private key (n,d) for decryption (d is secret); n is the product of two primes, while e and d mathematically derive from n [5]:



$n=p*q$ (p and q are prime numbers: they have only

Two divisors, 1 and itself);

$\phi(n) = (p-1)\ (q-1)$ is Euler's totient;

e coprime to $\phi(n)$;

$d*e \bmod \phi\ (n) = 1$;

C = Pe mod n (encryption);

P = Cd mod n (decryption);

where P is the plaintext and C the ciphertext. The encrypted message with RSA is a set of numerical values. These numbers will be converted using substitution in artificial DNA strand. All resulted peaces of DNA strands are bind together using a special ligase protein and the complementary strand as a template. The encrypted message can be transmitted in a compact form on DNA chip

B. Proposed Algorithm

In this paper we are proposing two algorithms

1) DNA cryptographic algorithm

2) Encrypt:

Step 1: data of which corresponding file converted to ASCII cod. For example original message:
"Requests" in ASCII will be:
114 101 113 117 101 115 116 115
Step 2: The converted ASCII codes of numeric values are arranged in a string and taken by several digits at once, number of digits rise together with the public keys length. Present example we will take seven digits at once and obtain:
114101 113117 101115 116115.
Step 3: These numbers, six digits long will be encrypted with public key and the result is another set of numbers:
417010496325959; 129526952115213; 373908236380170; 367569882589035.
Step 4: Encrypted sequence is transformed in binary form:
417010496325959→01011110111100101010101011111001110000110001111111.
Step 5: Binary sequence using substitution is transformed in DNA sequence:
A – 00
C – 01
G – 10
T – 11
010111101110001010101010111111001010000110001111111
→CCTGTGAGGGGGTTGCCAATATTTT
Step 6: All sequences are bind together in a single
strand, the cipher text:
CCTGTGAGGGGGTTGCCAATATTTTCTCCCTAAG
TCGACTCGGTCCTTCCTCCCTAAGTCGACTCGGTCC

TTCCCCCAACAATCCACGCGACATGGCGCCCCAAC
AATCCACGCGACATGGCGCCATGCATCCATAGGTCC.

C. Decrypt

Decryption is a reverse process: the DNA strand is cleaved in original peaces using restriction enzymes and transformed in numerical values using the same substitution as for encryption. The last step of decryption is done using the private RSA key Md Algorithm: The algorithm takes as input a message of arbitrarylength and produces as output a 128-bit message digest. The input is processed in 512-bit blocks. The processing consists of the following steps

1) *Step 1: Appending padding bits.:* The massage is padded so that its length in bits is congruent to 448 modulo 512 the length of the padded message is 64 bits less than an integer multiple of 512 bits
2) *Step 2:* Append length. A 64-bit representation of the length in bits of the original message (before the padding) is appended to the result of step 1 (least significant byte first). If the original length is greater than 264, then only the low-order 64 bits of the length are used. Thus, field contains the length of the original message, modulo 264. The outcome of the first two steps yields a message that is an integer multiple of 512 bits in length. In figure below, expended message is represented as the sequence of 512-bit blocks YY Y 01 1 L ,, , … − , so that the total length of the expanded message is L × 512 bits. Equivalently, the result is a multiple of 16 32-bit words. Let M[ ] 0 1 …N – denote the words of the resulting message, with N an integer multiple of 16. Thus, N = L ×16.
3) *Step 3:* Initialize MD buffer.

A 128-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as four 32-bit registers (A, B, C, D). These registers are initialized to the following 32-bit integers (hexadecimal values):

A = 67452301
B = EFCDAB89
C = 98BADCFE
D = 10325476

These values are stored in little-endian format, which is the least significant byte of a word in the low-address byte position. As 32-bit strings, the initialization values (in hexadecimal) appears as follows:

Word A: 01 23 45 67
Word B: 89 AB CD EF
Word C: FE DC BA 98
Word D: 76 54 32 10

4) *Step 4*: Process message in 512-bit (16-word) blocks.

The heart of the algorithm is a compression algorithm that consists of four “rounds” of processing; this module is labeled HMD5. The four rounds have the similar structure, but each uses a different primitive logical function, referred to as F, G, H, and I in the specification. Each round takes as input the current 512-bit block being processed (Yq) and the 28-bit buffer value ABCD and updates the contents of the buffer. Each round also makes use of one-fourth of a 64-element table T[ ] 1 64 … , constructed from the sine function. The ith element of T, denoted T[i], has the value equal to the integer part of 232 × abs i (sin( )) , where I is in radians.

5) *Step 5:* Output. After all L 512-bit blocks have been processed, the output from the Lth stage is the 160-bit message digest.

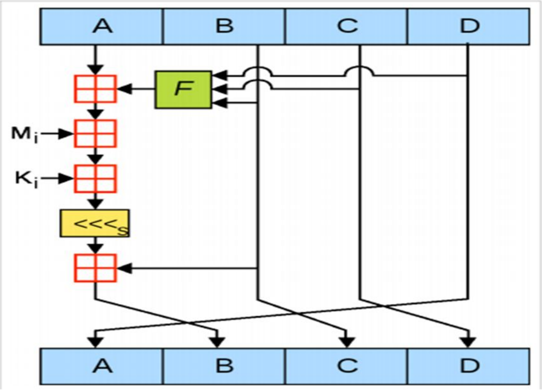
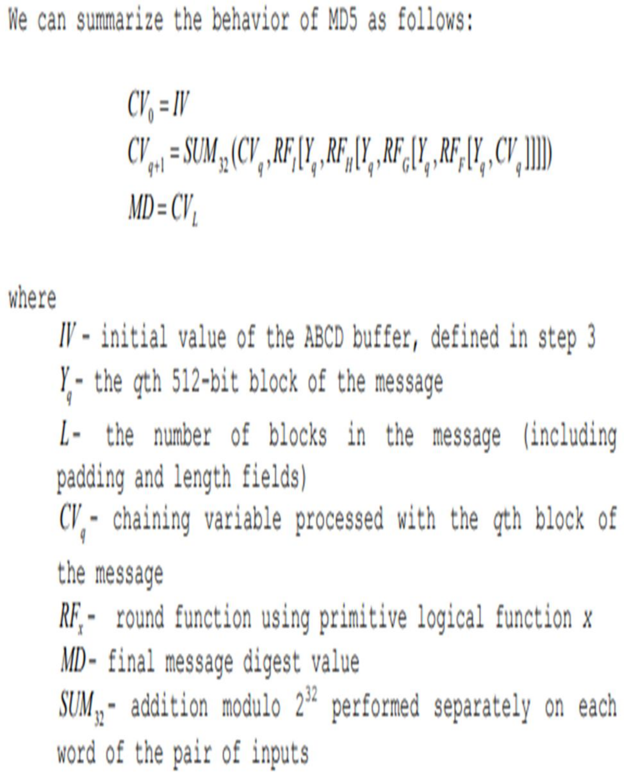


Fig2. Elementary MD5 operation



We can summarize the behavior of MD5 as follows:

$$CV_0 = IV$$
$$CV_{q+1} = SUM_{32}(CV_q, RF_I[Y_q, RF_H[Y_q, RF_G[Y_q, RF_F[Y_q, CV_q]]]])$$
$$MD = CV_L$$

where

IV - initial value of the ABCD buffer, defined in step 3

Y_q - the qth 512-bit block of the message

L - the number of blocks in the message (including padding and length fields)

CV_q - chaining variable processed with the qth block of the message

RF_x - round function using primitive logical function x

MD - final message digest value

SUM_{32} - addition modulo 2^{32} performed separately on each word of the pair of inputs

V. IMPLEMENTATION

we implement our system model with the pseudonym cryptography in hybrid cloud environment. As algorithm is constructed by java and jsp technology, the access request will be handled by the middleware chains, which can be used to estimate whether the access is valid or not. It provides interfaces to interact with users for CRUD operations on object/metadata. Besides the server side, system also consists of keystone component, which is responsible for identity management and Setup, Key Gen, Delegate implementations of authorities. When receives an access request, the middleware first reads token to get user label from keystone, then retrieves metadata server to get the object label. Then the access policy determination

is performed in middleware, If satisfied, the request would be sent to next middleware, otherwise it responds error code. In the system, we implement our scheme based on the pseudonym (MD5) and DNA-based cryptography library. But in industrial system, because user attributes are limited, the storage space of private key is very cheap. Meanwhile, the pseudonym key will be provide manually in any pseudonym systems like pen drive, if data consumer lost that pseudonym system anywhere, couldn’t access the application.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a pseudonym based cryptography for securing data and reduce the usage of cloud storage. We first formally introduced a DNA cryptographic key generation. Then, we proposed the deduplication scheme to reduce the usage of cloud storage. While processing deduplication file compressed to hash key. the hash key will serve as pseudonym key manually. The objects are encrypted with classification attributes and access policy, to achieve the constant-size cipher text components, which could be managed as the metadata. Therefore, both cipher text storage and communication cost are saved. The proposed scheme also has an advantage in distributed authorization to reduce the workload of system.

This paper is a significant work to address how to improve security of data by considering pseudonym based cryptography. DNA by considering the relationships among the attributes.

There are still several interesting open problems: How to Share pseudonym key to data consumer. We will focus on these problems in our future work.

REFERENCES

[1] L. M. Adleman. Molecular computation of solution to combinatorial problems. Science, 266:1021-1024, November 1994

[2] A. Gehani, T. LaBean, and J. Reif. DNA-Based Cryptography. Lecture Notes in Computer Science, Springer. 2004

[3] B. Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc, 1996

[4] H. Wang. Proving theorems by pattern recognition. Bell System Technical Journal 40, 1-42. 1961

[5] M. Factor, K. Meth, D. Naor, O. Rodeh, and J. Satran, "Objectstorage: The future building block for storage systems," in 2005 IEEE International Symposium on Mass Storage Systems and Technology.IEEE, 2005, pp. 119–123

[6] J. Li, Q. Wang, C. Wang, and K. Ren, "Enhancing attribute-based encryption with attribute hierarchy," Mobile networks and applications, vol. 16, no. 5, pp. 553–561, 2011

[7] Y. Wang, J. Yang, C. Xu, X. Ling, and Y. Yang, "Survey on access control technologies for cloud computing," Journal of Software, vol. 26, no. 5, p. 1129, 2015

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and communications security. Acm, 2006, pp. 89–98

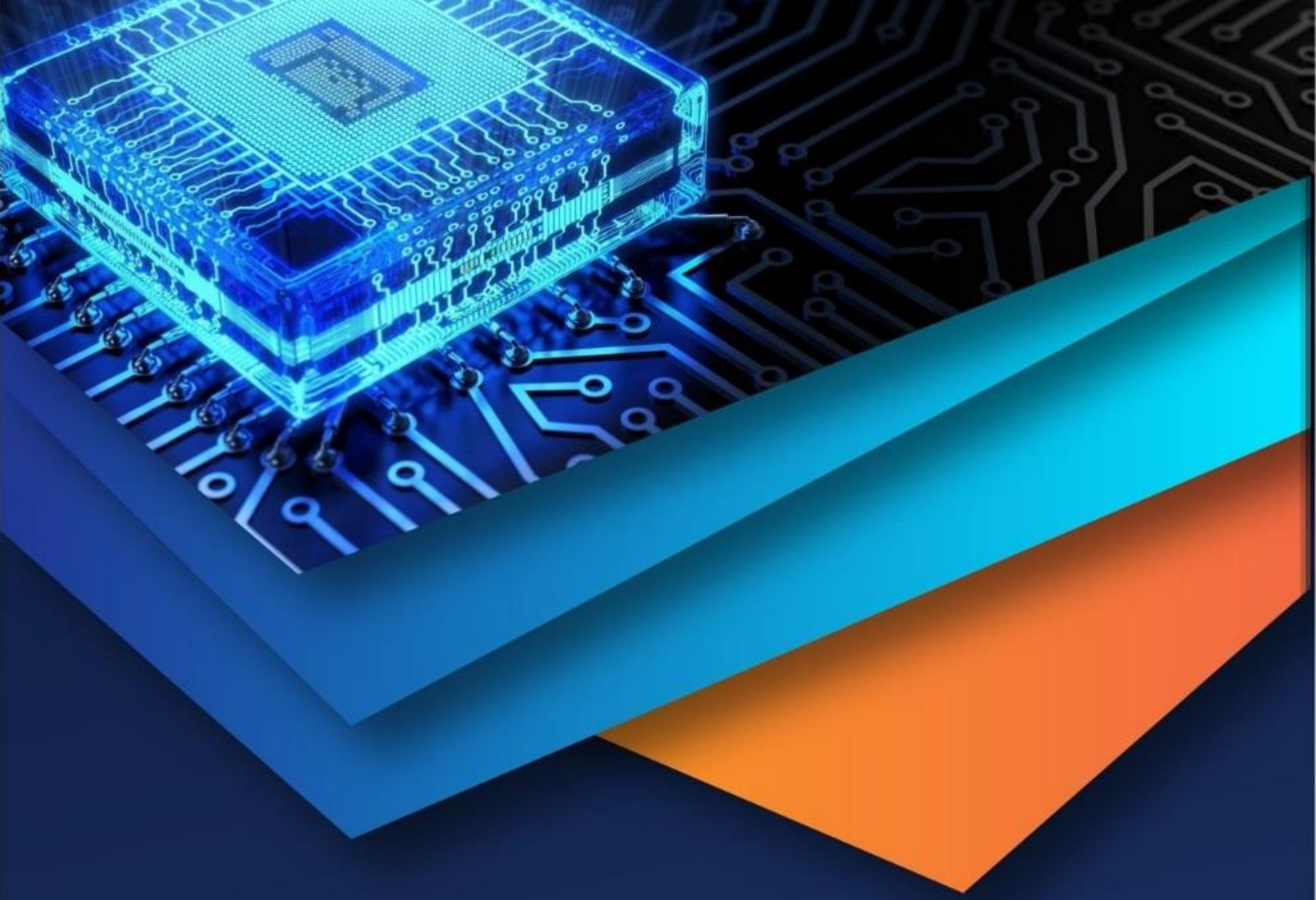
[9] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 1, pp. 131–143, Jan 2013

[10] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Annual International Cryptology Conference. Springer, 2001, pp. 213–229

[11] L. Cheung and C. Newport, "Provably secure ciphertext policy abe," in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 456–465

[12] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in International Workshop on Public Key Cryptography. Springer, 2011, pp. 53–70

[13] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-based data sharing scheme revisited in cloud computing," IEEE Transactions on Information Forensics and Security, vol. 11, no. 8, pp. 1661–1673,2016.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY