iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.36037

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

Secure Authentication in Cloud Computing using Diffie Hellman and One Time Password

Avleen Kour[1], Vibhakar Mansotra[2]
[1,2]*Department of Computer Science & IT, University of Jammu*

Abstract- Cloud Computing is the delivery of various computing services on Internet. These services include data storage, servers, databases, networking, software and analytics. Although cloud computing is a boon to the Information technology, there are certain issues which needs to be tackled. The biggest issue is that of cloud security. The user data needs to be protected from unknown parties. Different types of encryption schemes and authentication techniques are used to tackle the issue of cloud security. The encryption schemes are- Homomorphic Encryption and Homogeneous Encryption. In this paper, we have used a symmetric key algorithm- Diffie Hellman along with One time password (OTP) for authentication which gives more security to the user data.
Keywords- Cloud computing, FHE, Diffie-Hellman, OTP.

I. INTRODUCTION

The delivery of on-demand computing services over the web is known as Cloud Computing. These services range from application to storage and processing power. National Institute of Standards and Technology (NIST) defines cloud computing as the result of five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity or expansion and measured service. There are three service models and four deployment models as defined by NIST [1].

A. Service Models

There are mainly three service models for cloud computing as given below.

1) *Software as a service (SaaS):* It is also known as the on-demand software. It is a software distribution model in which a Cloud Service Provider hosts the applications. The applications can be accessed by the customers over the Internet. For example- Google apps, Drop box.

2) *Platform as a service (PaaS):* It is programming platform for programmers to build, test and run their applications. It gives the run-time environment to the users. The examples of PaaS are Google App Engine, Windows Azure, AWS Elastic Beanstalk.

3) *Infrastructure as a service (IaaS):* It is also called Hardware as a service. It delivers cloud computing infrastructure like server, storage, operating system and network. For example- Amazon Web Services (AWS), Google Computer Engine, Linode.

B. Deployment Models

There are four deployment models for cloud computing as given below.

1) *Private Cloud:* Private cloud is based solely for an organization. The resources are limited to the organization only. Private cloud is used when organizations want cost efficiency and greater control over their data, users and network.

2) *Community Cloud:* Community cloud is almost similar to private cloud. The only difference between a community cloud and a private cloud is that a community cloud shares the infrastructure of various organizations that have the same concerns like security, compliance, jurisdiction etc.

3) *Public Cloud:* Public clouds are accessible by the public over the internet. They are either paid or free of cost. Their infrastructure is also funded by the Cloud Service Providers.

4) *Hybrid Cloud:* Hybrid cloud can be defined as a combination of two or more different clouds. The cloud can in turn be any kind of cloud but it is all part of the same architecture.

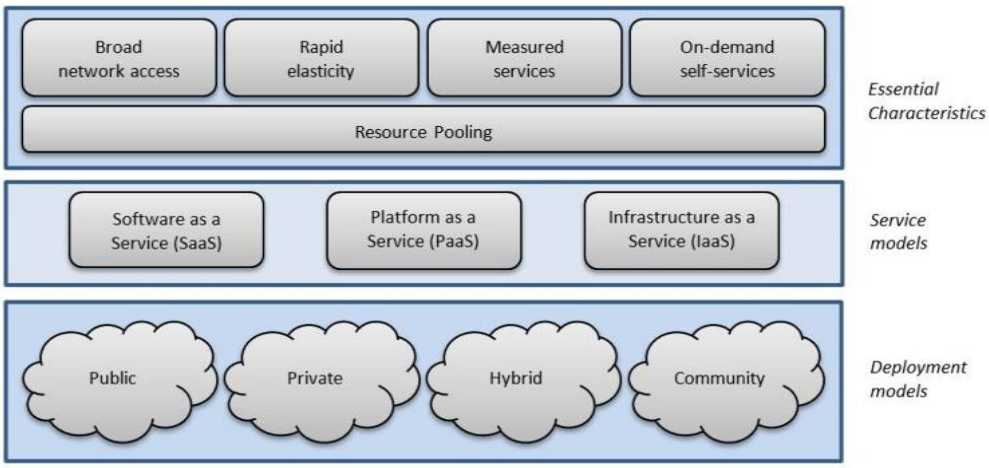


Fig 1: Fundamentals of Cloud Computing

C. Cloud Security

Cloud security can be defined precisely as the protection of data stored online via cloud computing platforms from unauthorized access, theft or deletion. Data protection is the biggest challenge posed by cloud computing [2]. To improve security in cloud computing, it is important to offer authorization, authentication and access control to data stored in the cloud [3]. There are three main components of cloud security as follows:

1) *Confidentiality:* To ensure data is protected from any attacks.
2) *Integrity:* To ensure that data is not lost or modified by any unauthorized access.
3) *Availability:* It depends on the agreement between the vendor and the client as to what data is accessible to the client to work on.

II. REVIEW OF LITERATURE

In recent years, cloud computing as a service has gained wide popularity. The major objective of this approach is to minimize the capital as well as operating costs of the system. Further, dynamic scaling is provided along with the deployment of new services which also does not require to maintain an infrastructure that is completely dedicated [4]. Thus, the manner in which the organizations look upon their IT resources has been transformed largely by the cloud computing applications [5]. The organizations today are adopting cloud computing instead of the single system that includes a single operating system as well as application. There are large number of resources available and it is possible for a user to select any number of resources required. At any interval of time, this on-demand service can be provided. All the important complex operations can be taken care of using CSP in these systems. Providing higher flexibility is the best advantage of cloud computing. In their paper, entitled "Secure cloud computing: Communication protocol for multithreaded homomorphic encryption for remote data processing", Alexander Oppermann et al [5], have worked on homomorphic scheme wherein the main challenge is key management and its subsequent sharing.

For providing secure connections in the cloud computing systems, there is a need to include several security mechanisms. Otherwise, the integrity of data can be lost at any time since an unauthorized user can have an access to this private data. To protect the information of a user within the cloud applications, several privacy techniques have been proposed earlier [6].

In their paper entitled "Enhanced Data Security in Cloud Computing with Third Party Auditor", Bhavna Makhija et al [7], have proposed various techniques along with their merits and demerits like Message Authentication Code (MAC) which protects the data and maintain its integrity. As per their paper, the owner of any information verifies the data integrity by recalculating the message authentication code of data received from others but recalculation is possible if the amount of data is very large. They have used hash trees for large files along with a third-party auditor which is used to relieve the large data into small parts of maintenance and security. The algorithm proposed by them focused on data integrity and dynamic data operations. The algorithm used encryption for ensuring the data integrity. Public key is also defined which is based on homomorphic authentication. For proof of retrievability, a hash function is used. This algorithm had a main drawback that it required implementation of the resources with higher cost.

In the paper entitled "Securing the cloud environment using OTP", Vimmi Pandey et all [8], have worked on Dynamic mobile token application. This is the application in mobile phones which is used to generate a code with the help of OTP (One Time Password). This OTP code can only be used once for login session. In this paper, one of the methods of OTP has been

described. They have used two phases in their paper, Registration phase and Login phase. User first registers itself by filling his/her credentials in the form and then enters the second phase, the Login phase. In login phase, OTP is generated by three parameters: the current time, 4-digit PIN code and Init-secret. This code is valid for three minutes only. This ensures protection against eavesdropper's attack and man-in-the-middle attack. Hence, they proved that using OTP is a very secure method.

In their paper entitled "Cloud Computing Security", Ankur Mishra et al [9], have worked on two techniques: Virtualization and Multi-tenancy which provides security about cloud computing. As data is organized by third party organizations, that offer SaaS and PaaS, Virtualization and Multi-tenancy techniques are used for the security purposes. Virtualization is a process of making a physical computer function as two or more computers whereas every other machine or computer is non-physical or virtualized. There are two types of virtualization namely, Full virtualization and Para virtualization and two architectures of virtualization called Hosted and Hypervisor architecture. Multi-tenancy is defined as the ability to provide computing services to multiple customers by using a common infrastructure and code base. Multi-tenancy can be applied to different levels that is, application level, middleware level, operating system, hardware level.

III. DIFFIE-HELLMAN AND OTP

From the literature review, we have seen that Fully Homomorphic encryption provides better security than full disk encryption (FDE). Unlike FDE, the encryption is not applied on full disk but the encryption is applied on each function. The cipher text and plain text is not related but the main emphasis is on the algebraic operation that works on both of them.

After the invention of RSA, Rivest, Adleman and Dertouzos introduced the idea of fully Homomorphic schemes. They worked on encryption functions that permitted encrypted data to be operated on without prior decryption of the operands and they called those schemes privacy homomorphisms [10].

Fully Homomorphic Encryption (FHE) can be used to for private queries in a search engine, not including, what is being searched. More accurately, FHE has many properties and applications. Assume that cipher text ci decrypt to plaintexts mi.

Therefore, Decrypt (ci) = mi

where the mi's and ci's are elements of some ring with two operations, addition and multiplication.

In FHE one has,

Decrypt (c1 + c2) = m1 + m2 and

Decrypt (c1 * c2) = m1 * m2

The operations are performed on encrypted data in fully homomorphic encryption and in this, private keys are not known. The secret key is only held by the client. When we decrypt the result of some operation, it is the similar as if we had approved the calculation on the raw data.

In this research, the fully homomorphic encryption scheme will be enhanced using Diffie Hellman and OTP technique which is defined as follows:

A. Diffie Hellman Key Exchange Algorithm

It was in 1976 that the first public-key algorithm: Diffie Hellman algorithm emerged within the public sphere. It's named after Whitfield Diffie and Martin Hellman. In this algorithm, sender and receiver make a common secret key and then they start communicating with each other over the public channel which is known to everyone [11]. The algorithm is as given below.

- Suppose two users A and B want to communicate to each other.
- They do not want their message to be known by the eavesdropper. Both the users A and B agree on and make public two numbers q and α, where q is a prime number and α is an integer which is primitive root of q. These numbers are publicly available. User A selects any random integer X_A, where $X_A<q$ and computes $Y_A= \alpha^{XA}$modq. User B also selects a random integer X_B, where $X_B<q$ and computes $Y_B= \alpha^{XB}$modq. For both the users, X is the private value whereas Y is the value available publicly.

User A	User B
Choose a secret number X_A. Compute $Y_A= \alpha^{XA}$(mod q)	Choose a secret number X_B. Compute $Y_B= \alpha^{XB}$(mod q)

Table 1- Private Computations.

- Thus, the public values generated are exchanged among both the users.
- User A sends Y_A to the User B.
- User B sends Y_B to the User A.
- User A calculates the secret key,

$k1=(Y_B)^{XA} \bmod q$

- User B calculates the secret key,
  $k2=(Y_A)^{XB} \bmod q$

If the keys k1 and k2 are equal, we say that the exchange between User A and B have been successful [12].

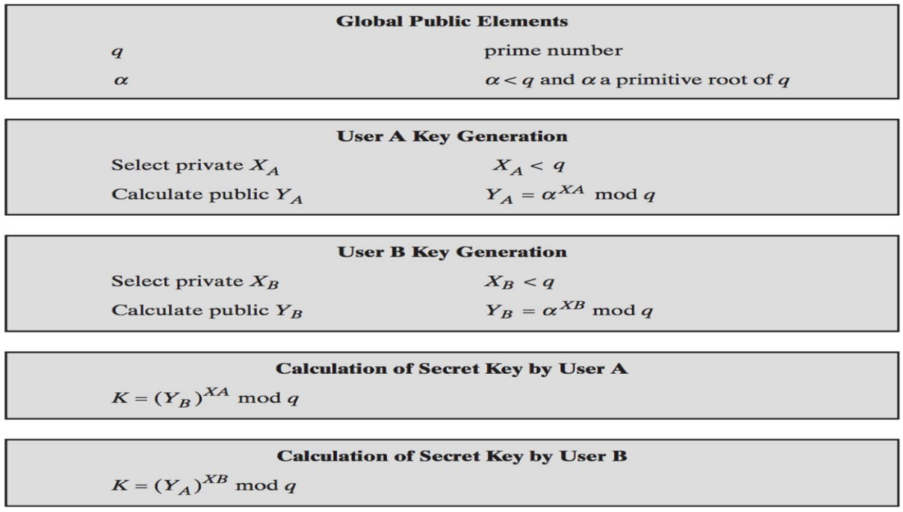


Fig. 2 Diffie Hellman key exchange algorithm

In the Diffie-Hellman algorithm, if two parties, for example User A and B wish to exchange information, both agree on a symmetric key. We use symmetric key for the encryption and decryption of the messages. Diffie-Hellman is used only for key agreement or key exchange and not for encryption and decryption. Therefore, before the communication has been started, a secure connection is established between both the users. Both the users select their own random number to establish a secure channel and thus, exchange the information.

B. One Time Password (OTP)

One time password or OTP is a password which is valid for one login session only. The most important feature of using OTP is that, unlike static passwords, OTP is not susceptible to replay attacks. When the secret key is generated by the Diffie-Hellman algorithm, the system will generate OTP based on the number of times the user gets login and the secret key. The OTP will be the (secret key+ OTP). If the user enters the correct OTP, he will thus be authenticated, otherwise it will get rejected.

IV. PROPOSED METHODOLOGY

For the problem of key sharing and key management in homomorphic encryption, an enhancement scheme has been proposed based on the fusion of Diffie-Hellman and OTP. The OTP is generated on the basis of the secret key which is generated from Diffie-Hellman algorithm. The algorithm is as given below:

1. Input: Data for encryption

2. Output: Encrypted Data

Logic

```
Key Generation ()

I=Input Data

For  I = 1 to it_Max

        For each particle p in P do

                Fp=f(p)

        If fp is better than f(pBest)
```

pBest=p;

end

end

gBest=best p in P

For each particle p in P do

V=V+Cl*rand*(pBest-p)+c8*rand*(gBest-p)

P=p+v

End

End

3. Key for Data encryption =P

4. If (user enter key=P)

Decrypt data;

Else

Display message wrong password

End

The user will enter the prime number and one random number for generating secret key on client side. Once the secret key is generated, the user will be asked to enter the OTP. If the correct OTP is entered, the encryption will take place or else it will get rejected with a message.

V. RESULTS

The whole scenario has been implemented on MATLAB tool where the following results are obtained. The data to be taken is in the form of images.

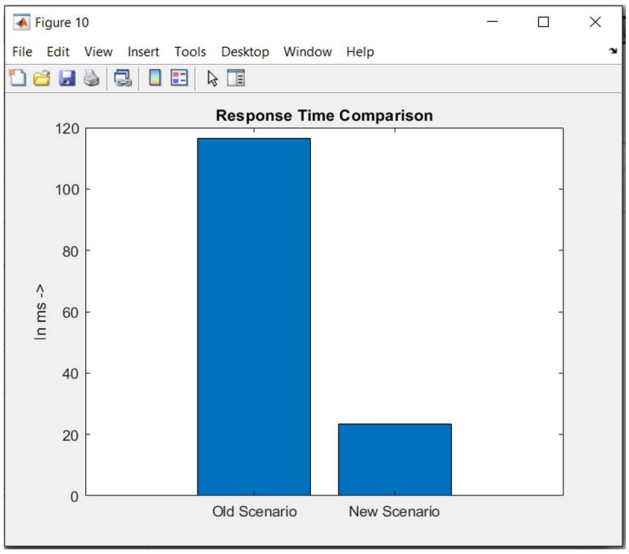


Fig. 3 Comparison graph in terms of response time

As shown in figure 2, the comparison between old and proposed scenario is given in terms of response time. The response time taken in the new scenario to encrypt an image is better as compared to the old scenario.

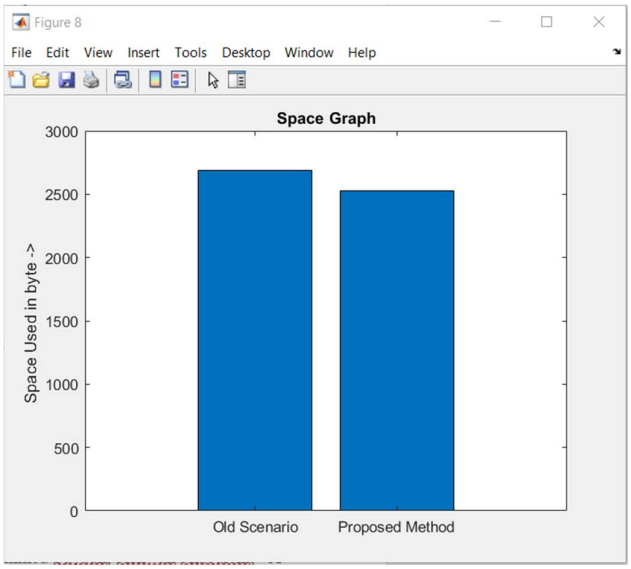


Fig. 4 Comparison graph in terms of bytes used

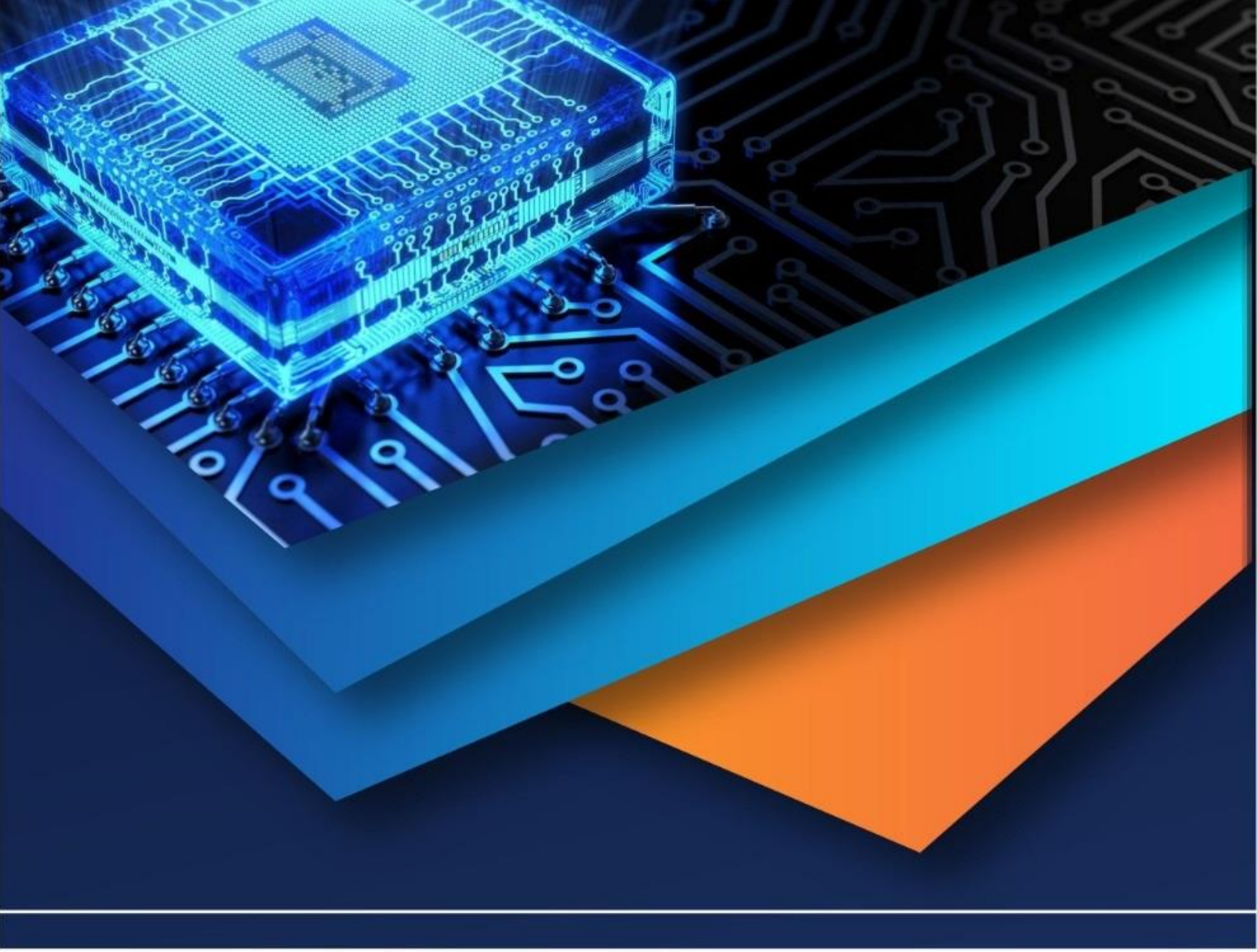
As shown in figure 3, the comparison between old and proposed method is given in terms of bytes used. The space consumed has decreased in the proposed method as compare to the old method. The data usage is less. Thus, we can say that the efficiency in terms of response time and space is better in the proposed model as compared to the previous models.

VI. CONCLUSION

Cloud computing and its security at the utmost is the need of the hour. In this work, we reviewed two main techniques for cloud data encryption. These are the full disk encryption and full homomorphic encryption. We find that the full homomorphic encryption technique is much better than the full disk encryption. But the main problem which full homomorphic encryption has is the problem of key sharing and key management. Therefore, we proposed a technique: Diffie Hellman key exchange algorithm along with OTP (One Time Password) to enhance the generation of the secret key. Each time the user communicates in the cloud, a new secret key is generated. This increases the security of the cloud thus, being more reliable and efficient from the previous systems. In future, this work can be further improved using other encryption techniques that can further lead to better time and space results.

REFERENCES

[1] https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp
[2] Elham Mohammed Thabit A Alsaadi, Sabah Mohammed Fayadh, Ashwak Alabaichi, "A Review on Security Challenges and Approaches in the Cloud Computing", 8th International Conference on Applied Science and Technology (ICAST), 2020.
[3] Syed Milad Dejamfar, Sara Najafzadeh, "Authentication Techniques in cloud computing: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 7, issue 1, pp. 995-99, 2017.
[4] Xidan Song, Y. Wang, "Homomorphic cloud computing scheme based on hybrid homomorphic encryption", 3rd IEEE International Conference on Computer and Communications, pp 2450–2453, 2017.
[5] Alexander Oppermann, Federico Grasso Toro, Artem Yurchenko, Jean-Pierre Seifert, "Secure cloud computing: Communication protocol for multithreaded fully homomorphic encryption for remote data processing", IEEE International Symposium on Parallel & Distributed Processing with Applications, pp. 503–510, 2017.
[6] Abhishek Mukherjee, Dhananjay Bisen, Praneet Saurabh, Lalit Kane, "Enhanced Homomorphic Encryption scheme with Particle Swarm Optimization for Encryption of Cloud Data", Springer, pp. 291-298, 2019.
[7] Bhavna Makhija, Vinit Kumar Gupta, "Enhanced Data Security in Cloud Computing with Third Party Auditor", International Journal of Advanced Research in Computer Science and Software Engineering, Issue 2, pp. 341-345, 2013.
[8] Vimmi Pandey, "Securing the Cloud Environment Using OTP", International Journal of Scientific Research in Computer Science and Engineering, Issue 4, pp. 38-43, 2013.
[9] Ankur Mishra, Ruchita Mathur, Shishir Jain, Jitendra Singh Rathore, "Cloud Computing Security", International Journal on Recent and Innovation Trends in Computing and Computation, vol. 1, issue 1, pp. 36-39, 2013.
[10] Ronald L., Rivest Len Adleman, Michael L. Dertouzos, "On data banks and privacy homomorphisms", Massachusetts Institute of Technology Cambridge, Massachusetts.
[11] Aryan, Chaithanya Kumar, Durai Raj Vincent P M, "Enhanced Diffie-Hellman algorithm for reliable key exchange", 14th International Conference on Science, Engineering and Technology, 2017.
[12] William Stallings, Cryptography and Network Security.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY