JCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

# Implementation on Shared Access Authentication in Cloud Computing Using Encryption Based Security Mechanism

**Ritu Pathak [1*], Lalit Chourasia [2], Rahul Sharma [3]**

[1]Computer Science, Babulal Tarabai Institute of Research & Technology, RGPV, Sagar, India
[2]Computer Science, Babulal Tarabai Institute of Research & Technology, RGPV, Sagar, India
[3]Computer Science, Babulal Tarabai Institute of Research & Technology, RGPV, Sagar, India

**Available online at: www.ijcseonline.org**

Accepted: 09/Jun/2018, Published: 30/Jun/2018

**Abstract**: Cloud computing is now a very common term among IT peoples. In cloud computing all the computing services like servers, networking devices, application is available over the Internet. Many people can share and access different application and resources as per their need and privileges of access & authorities. In cloud computing security & privacy of data is most important point which required lot of attention by the cloud service provider. Authentication is the main concern in existing security solutions, but neglect a subtle privacy issue during a user challenging the cloud server to request other users for data sharing. User's privacy is reveal by the challenged access request no matter it obtain the access permission or not. Ambition of this paper is to investigate and estimate the most important security techniques for data sheltering in cloud computing. We have proposed a Privacy Preserving Authentication Protocol based on Shared Authority (PPSA). Share access authority is achieved by the anonymous access request and analyzing and observing privacy. Data security is enhanced by using attribute base control system, which allows users to access their own data. To provide the data accessing from the other trusted party and sharing among the multiple users proxy re-encryption scheme is used by the cloud server. It indicates that the proposed scenario is possibly applied for enhanced privacy- preservation and security in cloud applications. We have done the review study on this scenario. In this paper we will focused on implementation of this review with assuring an improved result.

***Keyword:*** ECDH, SHA512, Authentication protocols, Cloud computing, Cloud computing security, Data security.

## I. INTRODUCTION

Cloud computing is the aptitude to access a group of computing resources owned and continued by a third party via the Internet. It is not a technology but a way of distribute computing possessions. Cloud computing usually involves the relocate, storage and dispensation of data. There are a lot of security techniques for data security that are acknowledged from the cloud computing providers, and they all make available verification, confidentiality, access control and agreement. Fig. 1 shows the system architecture in cloud computing environment. We propose Privacy- Preserving Authentication protocol
based on Shared Authority (PPSA) to concentrate on above privacy issue for cloud storage.

In the PPSA,

1) Shared access authority is accomplished by unidentified access demand corresponding mechanism with security and privacy contemplations (e.g., authentication, data ambiguity, user privacy, and forward security).

2) Attribute based access control is approved to appreciate that the user can only admittance its own data pasture;

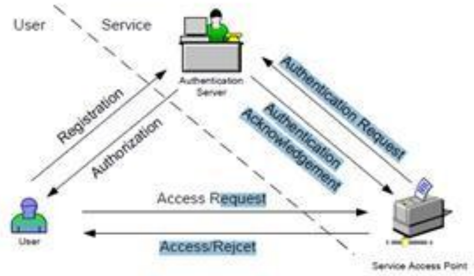


*Fig 1 System Architecture*

3) Proxy re-encryption is functional by the cloud server to supply data sharing along with the multiple users. Meanwhile, Universal Compos ability (UC) model is established to prove that the PPSA hypothetically has the plan correctness. It designated that the proposed protocol recognizing privacy-preserving data access

authority sharing, is attractive for more than one user mutual cloud applications.

Rest of the paper is organised as follows Section I contains the introduction of Cloud Computing and PPSA, Section II contains the related work, Section III contains the proposed method, Section IV contains the security algorithm used in my project, Section V contains the implementation and results discussion, Section VI concludes research work with future directions.

## 2. RELATED WORK

Hong Liu, Huansheng Ning, Qingxu Xiong and Lairence T. Yang[1] propose a Shared authority based Privacy - Preserving Authentication Protocol (SAPA) In SAPA Shared access authority is achieved by anonymous access request matching mechanism, attribute based access control is adopted to restrict the user access and data sharing is provided by proxy re-encryption. Antonis Michalas [2] propose Sharing in the Rain – a protocol based on Attribute-Based Encryption (ABE) which also remove access to a file without having to decrypt and re-encrypt the original data with a new key or a new policy. Ghassan O. Karame, Claudio soriente, Krzysztof Lichota, Srdjan Capkun [3] propose Bastion, a novel and efficient scheme that guarantees data confidentiality even if the encryption key is leaked and the adversary has access to almost all cipher text blocks. Jianghong Wei, Wenfen Liu, Xuexian [4] authors have propose a notion called revocable-storage identity-based encryption (RS-IBE), which can provide the forward/backward security of cipher text by simultaneously providing user revocation and cipher text update. Mazhar Ali, Saif U. R. Malik, Samee U. Khan [5] have propose data security for cloud environment with semi-trusted third party (DaSCE), a data security system that provides (a) key management (b) access control, and (c) file assured deletion. P. Vidhya L Lakshmi, Dr. S. Sankar Ganesh [6] has identified a new privacy challenge during data accessing. . Apurva Gomase, Prof.Vikrant Chole [7] proposed re-encryption in which the data is encrypting twice. Jingwei Li, dan Lin, Anna Cinzia Suicciarini, Jin Li, Chunfu Jia [8] propose a privacy-preserving STorage and REtrieval (STRE) mechanism that not only ensures security and privacy but also provides reliability guarantees for the outsourced searchable encrypted data.

## 3. PROPOSED METHOD

It is clear that the security issue has played the most important role in hindering Cloud computing acceptance. Without doubt, putting your data, running your software on someone else's hard disk using someone else's CPU appears daunting to many. While there is increasing use of cloud computing service in this new era, the security issues of the cloud computing become a challenges. Cloud computing must be safe and secure enough to ensure the privacy of the users.

In this paper we address all the privacy issue mentioned in introduction part to propose a privacy preserving authentication protocol based on shared authority for the cloud data storage, which realizes authentication and authorization without negotiating a user’s private and legal information. All the previous work in cloud storage mainly focuses on the authentication but neglect the privacy issue in which a user challenging the cloud server for data sharing where the challenged request itself reveal the user’s privacy no matter whether or not it can obtain the access authority. To accomplishing this goal we have proposed a Shared access authority is achieved by anonymous access request similar mechanism with security and privacy concerns (e.g., authentication, data anonymity, user privacy, and forward security).

Attribute based access control is accepted to realize that the user can only access its own data fields. Proxy re-encryption is applied by the cloud server to provide data sharing among the multiple users. This will specify the proposed protocol realizing privacy preserving data access authority sharing, and will be promises better multi-user collaborative cloud applications.

## 4. SECURITY ALGORITHMS USED IN CLOUD COMPUTING

### 1. ECDH( Elliptic Curve Diffi - Hellman)

Elliptic-curve Diffie–Hellman (ECDH) is an anonymous key agreement (In cryptography, a key-agreement protocol is a protocol whereby two or more parties can agree on a key in such a way that both influence the outcome). In this two parties, each having an elliptic-curve public–private key pair, establish a shared secret (A shared secret is a cryptographic key or data that is only known to the parties involved in a secured communication) over an insecure channel (In contrast to a secure channel, an insecure channel is unencrypted and may be subject to eavesdropping).This shared secret may be directly used as a key, or to derive another key (Such use may be expressed as DK = KDF(Key, Salt, Iterations) where DK is the derived key, KDF is the key derivation function, Key is the original key or password, Salt is a random number which acts as cryptographic salt, and Iterations refers to the number of iterations of a sub-function.). The key, or the derived key, can then be used to encrypt subsequent communications using a symmetric-key cipher (Symmetric ciphers use the same cryptographic keys for encryption of plaintext and decryption of cipher text.). It is a variant of the Diffie–Hellman (Diffie–Hellman key exchange (DH) is a method of securely exchanging cryptographic keys (A cryptographic key is a string of bits used by a cryptographic algorithm to transform plain text into cipher text or vice versa) over a public channel) protocol using elliptic-curve cryptography (Elliptic-curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to non-ECC cryptography**)**. ECDH is an analogous scheme based on addition of points on an elliptic curve. In this the basic operation is combined to create a primitive function known as a keyed one-way function. A keyed one-way function is a function that takes two inputs, one of which is private (e.g., the key), and produces one output. Given the two inputs, it must be straightforward to calculate the output. But, it must be computationally infeasible to calculate the key, using only the other input and the output. In this way, each party can use their private key without revealing it to anyone else, either the other party or an eavesdropper.

**Domain parameters**

Our elliptic curve algorithms will work in a cyclic subgroup of an elliptic curve over a finite field. Therefore, our algorithms will need the following parameters:
1. The prime p that specifies the size of the finite field.
2. The coefficients a and b of the elliptic curve equation.
3. The base point G that generates our subgroup.
4. The order n of the subgrouop.
5. The cofactor h of the subgroup.

**Algorithm**

Suppose two people, Alice and Bob, wish to exchange a secret key with each other. Following steps are used for doing this:

1. First, Alice and Bob generate their own private and public keys. We have the private key dA and the public key HA = dA G for Alice, and the keys dB and HB = dB G for Bob. Note that both Alice and Bob are using the same domain parameters: the same base point G on the same elliptic curve on the same finite field.

2. Alice and Bob exchange their public keys HA and HB over an insecure channel. The Man In the Middle would intercept HA and HB, but won't be able to find out neither dA nor dB without solving the discrete logarithm problem.

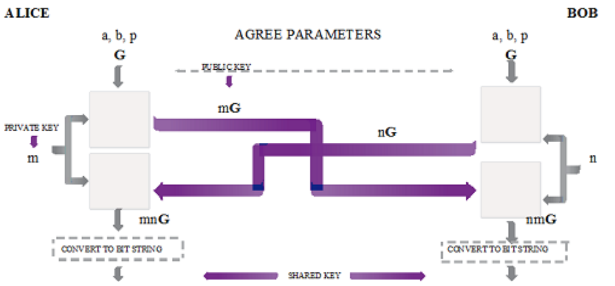


*Fig. 4.1 Key Exchange using ECDH*

3. Alice calculates S = dA HB(using her own private key and Bob's public key), and Bob calculates S = dB HA (using his own private key and Alice's public key). Note that S is the same for both Alice and Bob, in fact:
S = dA HB = dA (dB G) = dB (dAG) = dB HA

The man in the Middle, however, only knows HA and HB (together with the other domain parameters) and would not be able to find out the shared secret S.

**2. SHA (Secure Hashing Algorithm)**
Secure Hashing Algorithms, also known as SHA, are a family of cryptographic (cryptography is most often associated with scrambling plaintext into cipher text, then back again (known as decryption)) functions designed to keep data secured. It works by transforming the data using a hash function (A hash function is any function that can be used to map data of arbitrary size to data of a fixed size): an algorithm that consists of bitwise operations(a bitwise operation operates on one or more bit patterns or binary numerals at the level of their individual bits), modular additions( modular arithmetic is a system of arithmetic for integers, where numbers "wrap around" upon reaching a certain value), and compression functions(In cryptography, a one-way compression function is a function that transforms two fixed-length inputs into a fixed-length output). The hash function then produces a fixed size string that looks nothing like the original. These algorithms are designed to be one-way functions (function that is easy to compute on every input, but hard to invert given the image of a random input.), meaning that once they're transformed into their respective hash values, it's virtually impossible to transform them back into the original data. Here I'm using SHA 512 which belongs to SHA-2 family. SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic hash functions(a hash function which takes an input (or 'message') and returns a fixed-size alphanumeric string) designed by the United States National Security Agency (NSA)( is a national-level intelligence agency of the United States Department of Defense).They are built using the Merkle–Damgård structure(is a method of building collision-resistant cryptographic hash functions from collision-resistant one-

way compression functions), from a one-way compression function itself built using the Davies-Meyer structure (method to turn any normal block cipher into a one-way compression function )from a (classified) specialized block cipher. The SHA-512 Secure Hash Algorithm below figure shows the overall processing steps of SHA-512.

**Algorithm**

Here is a 4- step summary of SHA-512 processing:

STEP 1: Pad the message so that its length is an integral multiple of 1024 bits, the block size. The only complication here is that the last 128 bits of the last block must contain a value that is the length of the message.

STEP 2: Generate the message schedule required for processing a 1024-bit block of the input message. The message schedule consists of 80 64- bit words. The first 16 of these words are obtained directly from the 1024-bit message block. The rest of the words are obtained by applying permutation and mixing operations to the some of the previously generated words.

STEP 3: Apply round-based processing to each 1024-bit input message block. There are 80 rounds to be carried out for each message block. For this round-based processing, we first store the hash values calculated for the PREVIOUS MESSAGE BLOCK in temporary 64-bit variables denoted a, b, c, d, e, f, g, h. In the i th round, we permute the values stored in these eight variables and, with two of the variables, we mix in the message schedule word words[i] and a round constant K[i].

STEP 4: We update the hash values calculated for the PREVIOUS message block by adding to it the values in the temporary variables a, b, c, d, e, f, g, h.

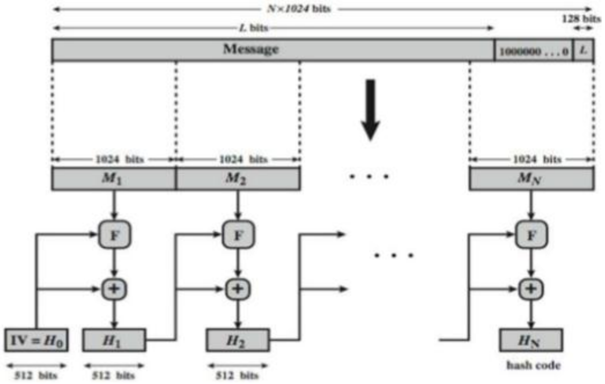
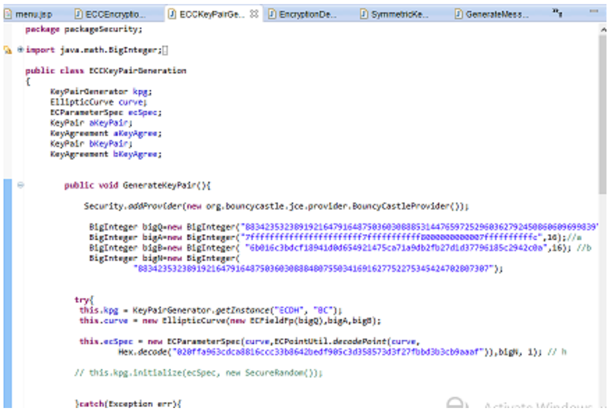


*Fig. 4.2 SHA – 512 processing of a single 1024 bit block*

## 5. IMPLEMENTATION AND RESULTS

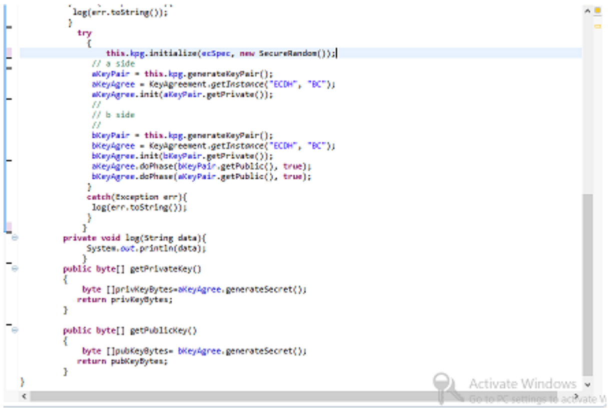
Implementation of algorithms has been done using Eclipse with Java. Codling's used for algorithms have shown below:

1. Coding used for ECDH



```
package packageSecurity;

import java.math.BigInteger;

public class ECCKeyPairGeneration
{
      KeyPairGenerator kpg;
      EllipticCurve curve;
      ECParameterSpec ecSpec;
      KeyPair aKeyPair;
      KeyAgreement aKeyAgree;
      KeyPair bKeyPair;
      KeyAgreement bKeyAgree;


         public void GenerateKeyPair(){

            Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider());

             BigInteger bigQ=new BigInteger("883423532389192164791648750360308885314476597252960362792450860609699839"
             BigInteger bigA=new BigInteger("7fffffffffffffffffffffff7fffffffffff8000000000007ffffffffffc",16);//a
             BigInteger bigB=new BigInteger( "6b016c3bdcf18941d0d654921475ca71a9db2fb27d1d37796185c2942c0a",16); //b
             BigInteger bigN=new BigInteger(
                     "883423532389192164791648750360308884807550341691627752275345424702807307");


          try{
           this.kpg = KeyPairGenerator.getInstance("ECDH", "BC");
           this.curve = new EllipticCurve(new ECFieldFp(bigQ),bigA,bigB);

           this.ecSpec = new ECParameterSpec(curve,ECPointUtil.decodePoint(curve,
                  Hex.decode("020ffa963cdca8816ccc33b8642bedf905c3d358573d3f27fbbd3b3cb9aaaf")),bigN, 1); // h

          // this.kpg.initialize(ecSpec, new SecureRandom());


          }catch(Exception err){
```

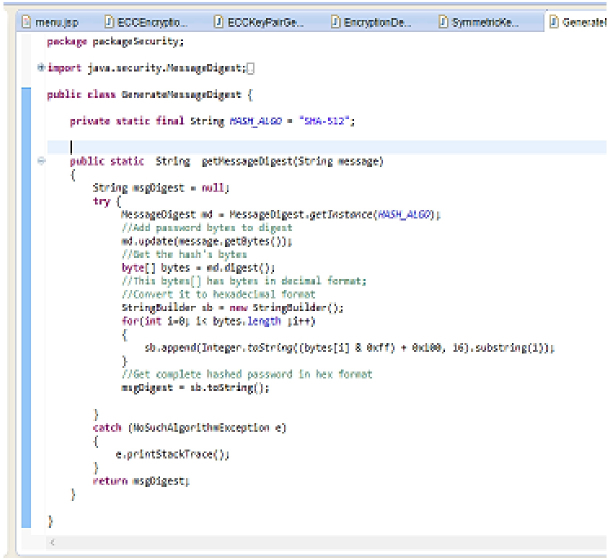
*Fig. 5.1 Coding of ECDH*



```
            log(err.toString());
          }
           try
            {
                 this.kpg.initialize(ecSpec, new SecureRandom());
             // a side
             aKeyPair = this.kpg.generateKeyPair();
             aKeyAgree = KeyAgreement.getInstance("ECDH", "BC");
             aKeyAgree.init(aKeyPair.getPrivate());
             //
             // b side
             //
             bKeyPair = this.kpg.generateKeyPair();
             bKeyAgree = KeyAgreement.getInstance("ECDH", "BC");
             bKeyAgree.init(bKeyPair.getPrivate());
             aKeyAgree.doPhase(bKeyPair.getPublic(), true);
             bKeyAgree.doPhase(aKeyPair.getPublic(), true);
            }
            catch(Exception err){
             log(err.toString());
            }
           }
        private void log(String data){
             System.out.println(data);
           }
        public byte[] getPrivateKey()
        {
           byte []privKeyBytes=aKeyAgree.generateSecret();
          return privKeyBytes;
        }

        public byte[] getPublicKey()
        {
           byte []pubKeyBytes= bKeyAgree.generateSecret();
          return pubKeyBytes;
        }
}
```

*Fig. 5.2 Coding of ECDH*

**2.** Coding used for SHA -512



```
package packageSecurity;

import java.security.MessageDigest;

public class GenerateMessageDigest {

    private static final String HASH_ALGO = "SHA-512";


    public static  String  getMessageDigest(String message)
    {
        String msgDigest = null;
        try {
             MessageDigest md = MessageDigest.getInstance(HASH_ALGO);
             //Add password bytes to digest
             md.update(message.getBytes());
             //Get the hash's bytes
             byte[] bytes = md.digest();
             //This bytes[] has bytes in decimal format;
             //Convert it to hexadecimal format
             StringBuilder sb = new StringBuilder();
             for(int i=0; i< bytes.length ;i++)
             {
                 sb.append(Integer.toString((bytes[i] & 0xff) + 0x100, 16).substring(1));
             }
             //Get complete hashed password in hex format
             msgDigest = sb.toString();

        }
        catch (NoSuchAlgorithmException e)
        {
            e.printStackTrace();
        }
        return msgDigest;
    }

}
```

*Fig 5.3 Coding for SHA - 512*

## 6. CONCLUSION

In this paper we have focused on giving access authority by keeping your privacy preserved in cloud computing world. Here we concede that during data retrieval reliability and privacy of data will be maintained by using authentication. Cloud server is reported user's access need, so for any unspecified access request cloud server will not entertain and user secrecy is enhanced. We have anticipated that attribute based access control will allow user to access its own data only and through proxy re- encryption cloud server can share data among multiple users by keeping privacy of user.

Thus for enhanced privacy conservation in cloud application the projected method can functional. As a future scope one can find an optimum and appropriate security solutions for the specific services in the Cloud. To concentrate on more specific areas like regulatory and compliance issues, jurisdiction laws, etc...

## 7. REFERENCES

[1]. Hong Liu, Huansheng Ning, Qingxu Xiong and Lairence T. Yang*," Shared Authority Based Privacy-Preserving Authentication protocol in Cloud Computing*" IEEE Transactions on Parallel and Distributed Systems ( Volume: 26, Issue: 1, Jan. 2015 )

[2]. Antonis Michalas, "*Sharing in the Rain: Secure and Efficient Data Sharing*" for the cloud" 11th International Conference or Internet Technology and Secured Transaction(ICITST)2016.

[3]. 3.Ghassan O. Karame, Claudio soriente, Krzysztof Lichota, Srdjan Capkun" *Securing Cloud Data under Key Exposure"* IEEE Transactions on Cloud Computing ( Volume: PP, Issue: 99 )Feb 2017.

[4]. 4.Jianghong Wei, Wenfen Liu, Xuexian Hu" *Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption".* IEEE Transactions on Cloud Computing ( Volume: PP, Issue: 99 )March 2016

[5]. 5. Mazhar Ali, Saif U. R. Malik, Samee U. Khan" *DaSCE: Data Security for Cloud Environment with Semi-Trusted Third Party*" IEEE Transactions on Cloud Computing ( Volume: 5, Issue: 4,)June 2015.

[6]. 6. P. Vidhya L Lakshmi, Dr. S. Sankar Ganesh, usi*"A secure cloud storage system with data forwarding ng proxy re-encryption scheme"*, international journal for trends in engg and tech , April 2015.

[7]. 7. Apurva Gomase, Prof.Vikrant Chole, "*Secure system implementation using attribute based encryption*", ijates,Vol.No.03,Special issue No.01,Nov 2015.

[8]. 8. Jingwei Li, dan Lin, Anna Cinzia Suicciarini, Jin Li, Chunfu Jia" *Towards Privacy-Preserving Storage and Retrieval in Multiple Clouds*" IEEE Transactions on Cloud Computing ( Volume: 5, Issue: 3, July-Sept. 1 2017 )October 2015.

[9]. 9. ] Deyanchen, hongzhao, "*Data security and privacy protection issue in cloud computing*"IEEE conference of computer science and electronics engg, 2013.

[10]. 10.Farheen Quazi & Prof. Vikrant Chole *"Implementation on Proxy-re-encryption based security mechanism to authenticate shared access in cloud computing*" International Journal of Scholarly Research (IJSR) Vol-1, Issue-2, 2017

[11]. 11. Archana Singh Parmar, Monika Sharma,"*Improving Data Storage Security in Cloud Computing using Elliptic Curve*" International Journal of Engineering Science and Computing(Vol -7,Issue- 4)April 2017.

[12]. 12. . Shweta Singh, Tabrez Nafis, Ankita Sethi,"*Cloud Computing: Security Issues & Solution*"International Journal of Computational Intelligence Research (Volume 13, Number 6, 2017).

[13]. 13. Balamurugan Balusamy, P. Venkata Krishna, G. S. Tamizh Arasi, and Victor Chang," *A Secured Access Control Technique for Cloud Computing Environment Using Attribute Based Hierarchical Structure and Token Granting System"* International Journal of Network Security (Vol.19, No.4, PP.559-572, July 2017)

[14]. 14. Mohamed Ali Hamza, Jianfei Sun, Xuyun Nie, Zhiquan Qin, and Hu Xiong," *Revocable ABE with Bounded Ciphertext in Cloud Computing*" International Journal of Network Security(Vol.19, issue:6)Nov 2017.

[15]. 15. Prabhleen Kaur Soul, Sunil Saini,"*data security approach in cloud computing using SHA*"international research journal of engineering and technology(Vol:4, Issue:6)June2017.

[16]. 16. P K Akulwar, R. V. Dharmadhikari, S. S. Turambekar, S. C. Dolli "*Cloud Computing: Data Storage Protocols and Security Techniques*" International Journal of Scientific Research in Computer Science and Engineering Volume: 6, Issue: 2, 2018

[17]. Sakshi kathuria " *A Survey on Security Provided by Multi-Clouds in Cloud Computing"* IJSRNSC Volume:6, Issue: 1, 2018