An Improved Authentication And Data Security Approach Over Cloud Environment

Rahul Sahu, Sanjana Rathour

Abstract: Cloud computing refers to the delivery of computing as a service rather than a product. Cloud computing is one of the most popular technology because cloud provides thousands of service at very lower price and we can use cloud service at anytime from anywhere only a network connection is required. The one of the best service of cloud computing is cloud storage. Cloud storage provides a facility to store data and share data among multiple users. In the previous work two factor data protection on cloud storage system is used to provide a secure data sharing among multiple users. They used device to store second key to perform second level decryption. The limitation with this approach is to carry additional hardware at all the places. Loosing of device may interrupt data access. We take this work as our initial work and decided to remove device dependency from this model. In our work we make a fully secure model that provides higher security during data sharing between users at cloud server. Additionally this approach helps in replacing device dependency using a secure model. Proposed approach helps in effective three factor security with low computational parameters. In our model we used two level encryption and one OTP based security that is the reason we are calling our model three factor security of data for cloud storage system. So, our approach is effective while looking at security aspect compare to previously one in cloud security.

Index Terms: Cloud security ,Two-way authentication, Storage services ,Key management ,key distribution, Elliptic Curve Cryptography, Blowfish Algorithm

————————————◆————————————

1 INTRODUCTION

Cloud computing is an era of computing it refers to the delivery of computing as a service rather than a product. Using cloud we can use the applications as utilities over the internet and it also allows user to create, configure and customize application online [1]. Cloud computing provides us a very important feature of storage through which we can store pool of data over cloud Figure1 which is mostly managed by third parties. Data is available on cloud can access from anywhere at any time only a strong network connection is required. The very important feature of cloud storage is sharing of data between multiple users [2]. Like a user stores his data on cloud and another user is able to access his data from cloud storage through network. As cloud storage has many advantages but the concern of cloud storage is to secure data from unauthorized access. When we share data over cloud that time some intruder can access our data and then misuses it. So to secure data we use the concept of encryption. Encryption is very essential concept in which we convert a readable message into unreadable format. Encryption is mainly divided into two categories (Asymmetric & Symmetric). In symmetric key encryption both the users use same key for encryption and decryption. In asymmetric encryption both the users use public key based encryption which depends on the identity of receiver. While the receiver uses the secret key to decrypt the message. In Asymmetric encryption a trusted third party is required which shares the combination of key between sender and receiver. In a cloud data storage system, the data owners would wish to specify the policies as to who can access their data and the cloud providers are required to correctly enforce the policies that the data owners have specified. In order to enforce the specified access control policies before putting the data onto the cloud, the data owners can encrypt the data in the way that only users that are wished by owner to allow as specified in the access control policies are able to decrypt and access the data. Several cryptographic schemes, such as the schemes in previous paper, have been developed to enforce access policies on outsourced data. These schemes combine cryptographic techniques and access control to protect the privacy of the data in an outsourced environment.

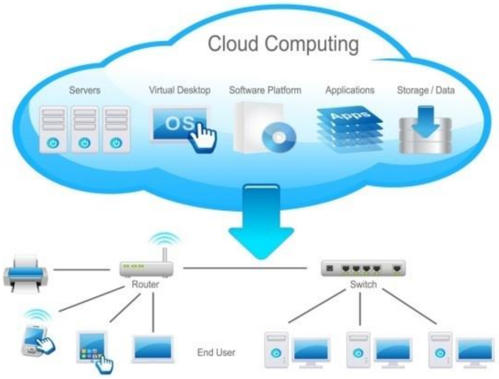

***Figure1 Cloud** Architecture*

1.1 Architecture

In the distributed computing architecture[1] of administration provisioning, fundamentally three gatherings are associated with giving administrations to the clients:-

1. User/Client
2. Third Party Auditor (TPA)
3. Cloud Server (CS) User/Client

In the distributed computing engineering, the user is the person who utilizes the administrations of the cloud. It might be a cell phone or stationary gadget which ask for administrations to the cloud specialist and after that on the premise of client's solicitations Third Party Auditor (TPA), gives requested administrations to these clients offered by the cloud Server (CS). In the distributed computing information is put away in serve from where information is gotten to when or wherever it is required. With the server, virtual servers are associated in which at least one virtual machines (VM) are arranged for calculation

- *Rahul Sahu is currenty Assistant Professor in Computer Sceince & engineering in LNCT Bhopal,India, , PH-9827337353. E-mail:rahulsahumanit@gmail.com*
- *Sanjana Rathour is currently pursuing masters degree program in Computer Sceince & engineering in LNCT Bhopal,India,PH-8962971740 E-mail rathoursanj28@gmail.com*

1.2 Third Party Auditor (TPA)

Outsider Auditor (TPA) is the outsider between cloud client and cloud specialist which are in charge of secure administration provisioning. In figure 2 we can see that how outsider reviewer giving administrations amongst clients and cloud servers safely. TPA plays out some operation for assention and confirmation amongst clients and specialist organization for security reason.

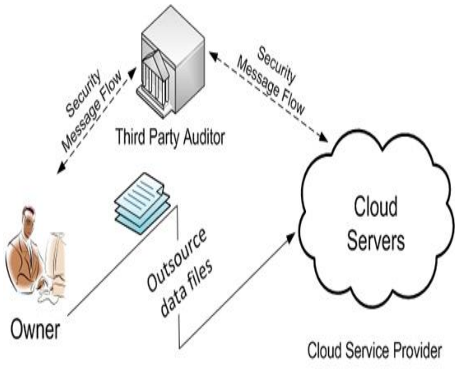

***figure 2 Cloud** computing TPA service provisioning architecture*

1.3 Our Contribution

- The significant element of our work is to remove device dependency second level encryption, so there is no requirement for repudiation in our work that spares hug amount of time.
- We utilize two level encryptions with IBE (Identity-based encryption) which means sender just utilizes the personality of the collector (name or Email-Id) no other data beneficiary is required.
- In our proposed work we present a significant component of email-based One Time Password (OTP) framework to make our framework progressively secure and adaptable.
- In our system, we use hashing procedure for second level decoding rather than gadget security, so we don't need of denial that makes our framework quicker and adaptable.

2 RELATED WORK

Here we examine some related work with our proposed work, and we will clarify why they are bad and adaptable as of our system.

2.1 Two-Factor Security Using Two Secret Keys

Two-level encryption primarily dependent on two type first certificate less and second authentication based [3]. In authentication based framework, a client picks his open key with a mystery key and the specialist produce the fractional key dependent on the character of the client. Encryption or mark check the two needs an open key and client character. At the recipient end unscrambling or signature check requires both fractional key and mystery key. So this framework requires exorbitant endorsement approval process. In declaration less [4], encryption we evacuate expensive approval yet at the same time it isn't advantageous it doesn't pursue character based encryption.

2.2 Two-Factor Security with Online Authority

In this approach, an online alters or is required for the safe exchange. The online go between is known as SEM (Security Mediator) [5, 6] and it gives security capacities. In this methodology, each exchange is reliant on SEM. On the off chance that SEM is repudiated somebody, at that point that client can't get the message. So in SEM framework if the network is not good then also we suffer to complete the transaction.

2.3 Two-Factor Security with Security Device

Two-way security verification is a significant part of information stockpiling and access [7, 8]. In this framework, two-level encryption has played out the PKG gives the key for first level encryption to the sender and sends a mystery key to the beneficiary for first level unscrambling and after that the recipient transfers the document at the cloud. For second level encryption, SDI gives another key for second level unscrambling, and it sends the decoding key to beneficiary's security gadget. Without either key, the message didn't get decoded. The one of the huge worry of this framework is in the event that the gadget is taken or lost, at that point we can't recover the message. This issue is settled in next work that is two-factor protections with denial.

2.4 Two-Factor Security with Security Device with Revocation

This system is same as the past system; however it has one extra element called renouncement [9, 10]. It means when the device is taken or lost then the recipient makes an impression on SDI and after that SDI refreshes the past calculation and produces another security gadget for unscrambling of the message. This framework is by all accounts great, however it takes two fold time if the device is lost that is the explanation we make free device system.

3 PROPOSED ALGORITHM OVERVIEW

We are using two different encryption technologies one is identity based encryption and another is Symmetric Key Encryption. At the first step PKG generate private key and it gives it to Identity Based Encryption(IBE) algorithm to perform first level encryption using ECC algorithm. Then we share this private key using Diffie-Hellman key exchange algorithm. Then we upload first cipher text C1 at cloud server. Then we perform second level encryption at cloud server. So before performing second level encryption we need a key, we share keys between sender and receiver using email based OTP system. So after getting key we use Blowfish algorithm at second level encryption. So, we are using three main very efficient and flexible algorithms ECC, Diffie-Hellman, Email based OTP and Blowfish.

3.1 Elliptic Curve Cryptography

ECC makes keys through the properties of the elliptic curve condition instead of the customary system for age as the aftereffect of extremely enormous prime numbers. The advancement can be used identified with most open key encryption techniques, for instance, RSA, and Diffie-Hellman. According to a couple of

authorities, ECC can yield a level of security with a 164-piece key that various systems require a 1,024-piece key to achieve.
The condition of an elliptic bend is given as,

$$y^2 = x^3 + ax + b$$

Few terms that will be used,

E = Elliptic Curve
P = Point on the curve
n = Maximum limit (This should be a prime number)

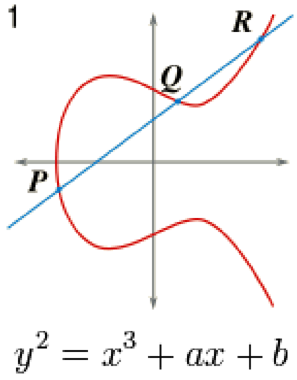

Figure 3. *Simple Elliptic Curve*

3.2 Blowfish Algorithm

Blowfish is an encryption calculation that can be utilized as a substitution for the DES or IDE Algorithms. It is a symmetric (that is, a mystery or private key) square figure that uses a variable-length key, from 32 bits to 448 bits, making it helpful for both local and exportable use. (The U. S. government denies the exportation of encryption programming utilizing keys bigger than 40 bits aside from in extraordinary cases.) Blowfish was structured in 1993 by Bruce Schneier as an option in contrast to existing encryption calculations.

3.3 Blowfish Algorithm Pseudo Code

```
uintA P[18];
uintB S[4][256];

uintA f (uintA x) {
  uintA h = S[0][x >> 24] + S[1][x >> 16 & 0xff];
  return ( h ^ S[2][x >> 8 & 0xff] ) + S[3][x & 0xff];
}
void encrypt (uintA & L, uintB & R) {
int i=0;
  while( i<16) {
    L =L^ P[i];
    R =R^ f(L);
    R = R^P[i+1];
    L =L^ f(R);
    i=+ 2;
  }
  L =L^ P[16];
  R =R^ P[17];
  swap (L, R);
}
void decrypt (uintA & L, uintB & R) {
int i=16;
  while(i > 0 ; ) {
    L = L ^P(i+1);
    R =R ^ f(L);
    R = R^P(i);
    L= L^f(R);
    i=i+2;
  }
  L =L^ P[1];
  R =R^ P[0];
  swap (L, R);
}
```

Initializing the P-array and S-boxes with values derived from pi; omitted in the example

```
{
int j=0;
while(j<18)
{
    P[i] = P[i] ^ key[i % keylen];
    ++j;
}
  uintA L = 0, R = 0;
int j=0;
  while ( i<18) {
    encrypt (L, R);
    P[i] = L; P[i+1] = R;
    i=i+2;
  }
  for (int i=0 ; i<4 ; ++i)
{
Int j=0;
    while(j<256) {
      encrypt (L, R);
      S[i][j] = L; S[i][j+1] = R;
j=j+2;
    }
}
```

4 PROPOSED WORK

In this work we proposed a model which covered all the loophole of previous work. So here we give an intuition on all the work.
We have some entities in our work which we explained in details.

- Private Key Generator (PKG1): It is a trusted third party which generates a secret key for first level encryption.
- Private Key Generator (PKG2): It is a trusted third party which generates a secret key for second level encryption
- Sender (Alice): Sender sends the data to the receiver, he only knows the identity (email-Id) of the receiver no other information of the receiver is required. It uploads the data on cloud after first level encryption.
- Receiver (Bob): Every receiver has a unique identity (email-Id or name), the receiver first gets the OTP on his email after successful submission of OTP, the receiver can login into the cloud server, and he can download the file from the cloud server and then perform two-level decryption on it.
- Cloud server: Cloud server plays a very important role in our system. The server stores cipher text which is downloaded by the receiver.

4.1 Steps performed at Sender End :

1. Generate key: First we give collector's email-id as a identity to PKG (Private Key Generator). At that point the PKG will give the blend of open key and private key to perform encryption and unscrambling. Open key is utilized by sender to encode the first message.
2. Select the file: In this phase we select the document which we need to send the receiver and afterward offer it to ECC calculation as an info which convert it into emblematic structure
3. First level encryption: Here we perform encryption by ECC calculation to encode the message. We give the chose record and open key as a contribution to ECC calculation and get the encoded document (C1) as a yield. At that point we transfer the document at cloud server
4. Generate the Hash value: we perform encryption by ECC calculation to encode the message. We give the picked record and open key as a commitment to ECC estimation and get the encoded archive (C1) as a yield. By then we move the record at cloud server
5. Second level encryption: Here, we perform second level unscrambling through Blowfish calculation. We give the C1 (scrambled record) and the hash esteem (SHkey) to Blowfish as info and after that get the encoded document C2 as yield.

4.2 Steps performed at Receiver End :

1. Generate the OTP: In the very first step at receiver side the collector gets the OTP on his email id. Thus, when the collector enters the privilege OTP then no one but he can login to get the document sent by the sender. In the event that we utilize versatile based OTP method, at that point our model would be gadget free. In this way, to evacuate gadget reliance we use email based OTP idea.
2. Generate the hash value: Here the beneficiary gets his own examining subtleties from Data Manager (DM) and apply SHA-2. on it to produce same SHkey to perform first level decoding.
3. Perform first level decryption: In this step, we perform Blowfish decoding calculation to unscramble the message. We give C2 and SHkey as contribution to Blowfish calculation and after that get the unscrambled document C1 as yield.
4. Second level decryption: After first level decryption we perform second level decoding through ECC unscrambling calculation. We give c1 and private key as contribution to ECC calculation and get the first document as yield.
5. Get the original data: After performing first and second level decryption we get the original data sent by the sender

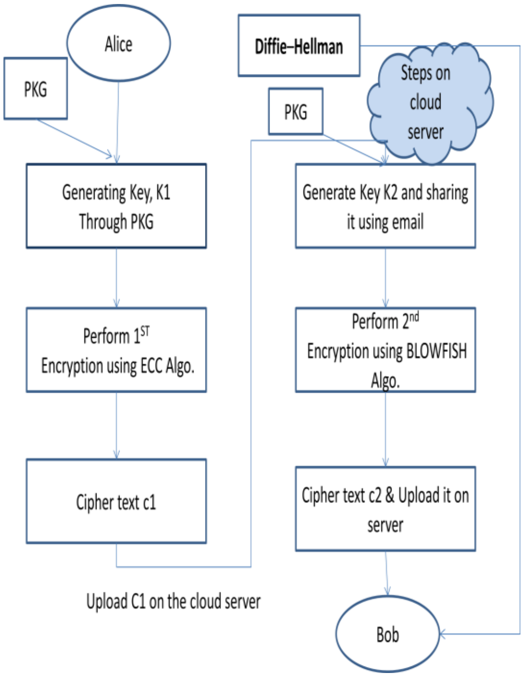
Alice
PKG
Diffie–Hellman
Steps on cloud server
PKG
Generating Key, K1 Through PKG
Generate Key K2 and sharing it using email
Perform 1ST Encryption using ECC Algo.
Perform 2nd Encryption using BLOWFISH Algo.
Cipher text c1
Cipher text c2 & Upload it on server
Upload C1 on the cloud server
Bob

***Figure 4.1**Sender side cloud model*

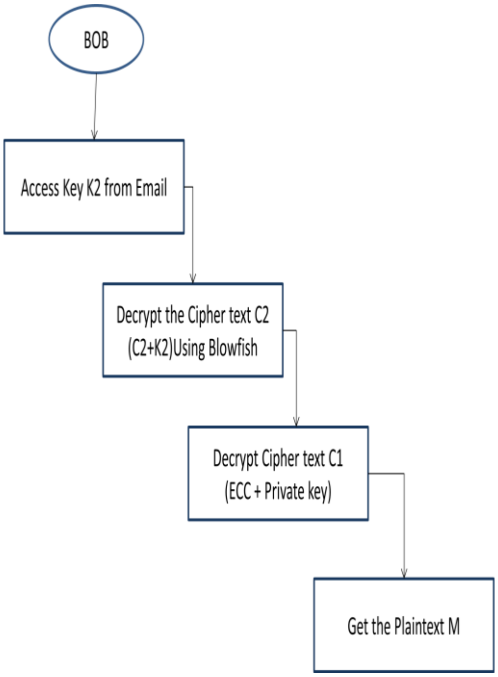
BOB
Access Key K2 from Email
Decrypt the Cipher text C2 (C2+K2)Using Blowfish
Decrypt Cipher text C1 (ECC + Private key)
Get the Plaintext M

***Figure 4.2** Receiver side cloud model.*

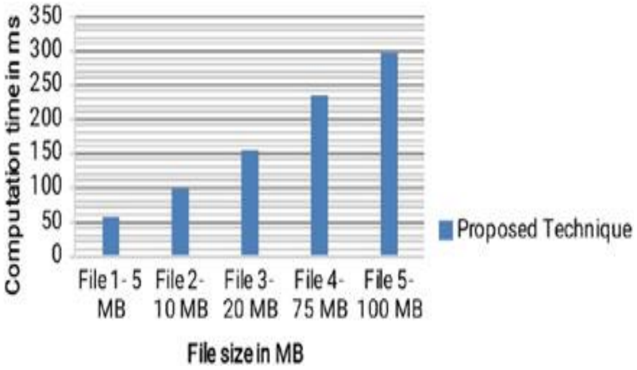

Fig. 4.3 *Comparison graph analysis of the proposed approach*

5 CONCLUSION

In this paper, we build up an interface which is furnishing three-level security with secure email-based OTP system. In our proposed work, we center to make truly adaptable and dependable interface through which anybody can share information over cloud server. In this paper, they give two-level security, yet the escape clause in that paper was the interface was reliant on the gadget that makes framework exorbitant and tedious. Along these lines, our fundamental point is to make a gadget free framework with exceptionally high-security information sharing methodology.

REFERENCES

[1] Sahai, Amit, Hakan Seyalioglu, and Brent Waters. "Dynamic credentials and ciphertext delegation for attribute-based encryption." In Annual Cryptology Conference, pp. 199-217. Springer, Berlin, Heidelberg, 2012.

[2] Chen, Henry CH, Yuchong Hu, Patrick PC Lee, and Yang Tang. "NCCloud: A network-coding-based storage system in a cloud-of-clouds." IEEE Transactions on computers 63, no. 1 (2014): 31-44.

[3] Liu, Joseph K., and Jianying Zhou. "Efficient certificate-based encryption in the standard model." In International Conference on Security and Cryptography for Networks, pp. 144-155. Springer, Berlin, Heidelberg, 2008.

[4] Hwang, Yong Ho, Joseph K. Liu, and Sherman SM Chow. "Certificateless Public Key Encryption Secure against Malicious KGC Attacks in the Standard Model." J. UCS 14, no. 3 (2008): 463-480.)

[5] Hwang, Yong Ho, Joseph K. Liu, and Sherman SM Chow. "Certificateless Public Key Encryption Secure against Malicious KGC Attacks in the Standard Model." J. UCS 14, no. 3 (2008): 463-480.

[6] Yap, Wun-She, Sherman SM Chow, Swee-Huay Heng, and Bok-Min Goi. "Security mediated certificateless signatures." In Applied Cryptography and Network Security, pp. 459-477. Springer, Berlin, Heidelberg, 2007.

[7] Singhal, Manav, and Shashikala Tapaswi. "Software tokens based two factor authentication scheme." International Journal of Information and Electronics Engineering 2, no. 3 (2012): 383-386.

[8] Sanka, Sunil, Chittaranjan Hota, and Muttukrishnan Rajarajan. "Secure data access in cloud computing." In 2010 IEEE 4th International Conference on Internet Multimedia Services Architecture and Application, pp. 1-6. IEEE, 2010.

[9] Liu, Joseph K., Kaitai Liang, Willy Susilo, Jianghua Liu, and Yang Xiang. "Two-factor data security protection mechanism for cloud storage system." IEEE Transactions on Computers65, no. 6 (2016): 1992-2004.

[10] Seo, Jae Hong, and Keita Emura. "Efficient delegation of key generation and revocation functionalities in identity-based encryption." In Cryptographers' Track at the RSA Conference, pp. 343-358. Springer, Berlin, Heidelberg, 2013.