# AUDIT-FREE CLOUD STORAGE VIA RANDOMIZED KEY PROTOCOL

**H.Anila Glory,**
Assistant Professor,
Department of Computer Science and Engineering,
Velammal Institute Of Technology,
Chennai,India.

**P.S.Revathy,**
B.E Student,
Department of Computer Science and Engineering,
Velammal Institute Of Technology,
Chennai,India.

**S.Anusha,**
B.E Student,
Department of Computer Science and Engineering,
Velammal Institute Of Technology,
Chennai,India.

**G.Sandhya,**
B.E Student,
Department of Computer Science and Engineering,
Velammal Institute Of Technology,
Chennai,India.

**Abstract**—Even though the technology faces several significant challenges, many vendors and industry observers predict a bright future for cloud computing. Cloud computing refers to the delivery of computing resources over the Internet. Because of its privacy, many encryption scheme has been proposed. The encryption process is converting electronic data into another form, called cipher text, which cannot be easily understood by anyone except the authorized parties. Still there are ways where the data can be decrypted back to original text. In this paper, we introduce a new scheme for the security in cloud using key protocol. This system generates a key for every user who have access to the file on the cloud server. Since this system is based on key protocol, the coercers will not be able to attack the file thus maintaining a high security.

***Keywords: Randomized key protocol, Attribute-Based Encryption, Cloud Storage, cipher text.***

## I. INTRODUCTION

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. Another potential benefit is that information may be better protected in the cloud. Predominantly, while there are benefits, there are privacy and security concerns too. The current system is based on Deniable Attribute Based Encryption scheme where encryption and decryption methodology is used. Deniable encryption involves senders and receivers creating convincing fake evidence of forged data in cipher texts such that outside coercers are satisfied. The attribute based encryption uses an attribute for the key that is generated. We make use of this idea, such that cloud storage providers can provide audit-free storage services. In the cloud storage scenario, data owners who store their data on the cloud are just like senders in the deniable encryption scheme. Those who can access the encrypted data play the role of receiver in the deniable encryption scheme, including the cloud storage providers themselves, who have system wide secrets and must be able to decrypt all encrypted data [1]. In our approach, the key distributor generates a unique key for the file to the user who has access to the respected files. When a user tries to access the file which has not been permitted for that user, the cloud admin marks him as the attacker. Thus maintaining privacy and security at high standards.

## II. EXISTING SYSTEM

Security is most vital part of cloud computing. Cloud computing faces many of the same threats as the physical workplace.

The existing systems that were used before fuzzy-Identity Based Encryption, Fine grained access control and Cipher text Attribute Based Encryption. In fuzzy Identity Based Encryption, A Fuzzy IBE scheme uses a private key for an identity, A, to decrypt a cipher text encrypted with an identity, B, if and only if the identities A and B are close to each other as measured by the "set overlap" distance metric. Therefore, this system allows for a certain amount of error-tolerance in the identities. This concept of Fuzzy Identity Based Encryption, which allows for error-tolerance between the identity of a private key and the public key used to encrypt a cipher text. However the fuzzy logic is effective, it has some cons. This system provides the user with an audited file, making it effectiveness by auditing the files and data that has been uploaded onto the cloud server. This reduces the security level of the system. In this system the basic construction does not use the random oracles and provides public access to the data owner. It gives access to only the selected ID. In the current system, i.e Deniable Attribute-based Encryption uses the encryption method which has various attributes tagged with the key generated. The hackers can determine the type of key used for privacy of data's in the cloud. These keys are finite in length and thus can be determined by diverse combinations. This is a hindrance for the security of the data's over the server. To overcome this problem of security, we propose a scheme where a unique key will be generated for each user and data that will be uploaded on the cloud server.

## III. PROPOSED METHODOLOGY

At present everyone opts for cloud storage as it provides high standard of security and storage space. Right now, there are many algorithms and encryption schemes in

*IJCRCST © 2016 | All Rights Reserved*

practice. The most frequently used methods are fuzzy logic, Attribute Based Encryption using public finite key. But this does not provide security at high standards, the use of public key will allow users to get the data of other users. To overcome this con of the existing system, we move for the use of the randomized key protocol. Here the methodology is, when the data owner uploads the data, it directly gets stored in the cloud server. When the user wants to retrieve the file or data, he requests for a secret key. The Key distributor generates the key for each file on the cloud server. This key is unique for each user. When a user tries to get the data of other user, the cloud admin will mark them as the hacker and will block the access for that user. Thus, preventing unauthorized access by the users. We propose this system to be highly secured because of the use of the Randomized key. The key generated will be an infinite key, i.e. there is no finite length for the key. The length of the key can be modified frequently, making it difficult to get the combinations. This is the reason that this method is difficult to hack. The algorithm used in this system is the "Blowfish" algorithm. We go for this algorithm because it supports all types of file extension.

## IV. ALGORITHMS

Encryption/decryption algorithm--blowfish: Blowfish is a symmetric block cipher algorithm for encryption/decryption. Blowfish is accepted as a fast and strong encryption algorithm because it has not been cracked. In our proposed system we use this algorithm because, it can be used for files with diverse extensions. Since we opt for the random key generation, it will be highly secured because the hackers cannot predict the type of algorithm used for the encryption process. The advantages of blowfish algorithm are that it is secure and easy to implement and best for hardware implementation.

## V. ARCHITECTURE DIAGRAM

There are four modules; Data owner, Key admin, Cloud admin, Users. The key admin is the one which generates the unique key for each user's data on the server. When the data owner uploads the file it will be encrypted and can be accessed only by providing the secret key.

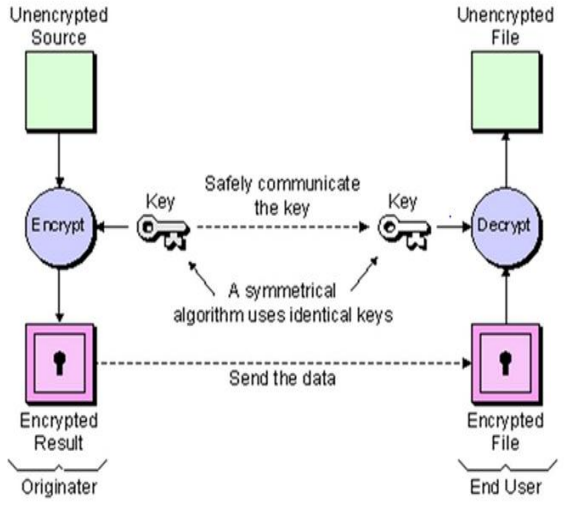


**Figure1:ArchitectureDiagram**

## VI. MODULE DESCRIPTIONS

The various modules developed in our proposed method involves the following:

- Data Owner
- Key Distributor
- Cloud Admin
- Users

**Data Owner:** Data Owner uploads files to group users via Cloud Storage like Drop box. If Owner Upload a file in background that the file will be encrypted using Blowfish Cryptography and validation the Time based one time password and enter in to cloud storage.

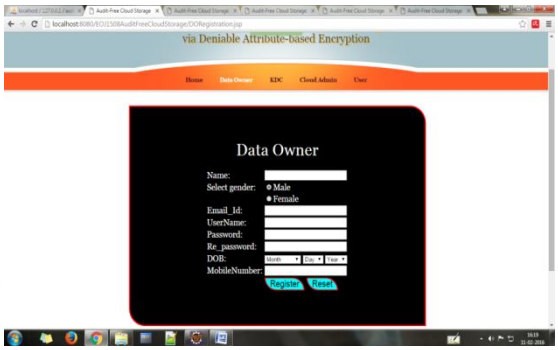


**Figure 2: Data Owner**

**Key Distributor:** Key Distributor send all the files decrypt key as well as the secret key to the users audits all the files present in cloud storage and check the users status , file status and attacker status in our application.

**Figure 3: Requesting Secret Key**

**Cloud Admin:** Cloud Admin Verify the Auditor approve files after grant the final approval and sends that files to third party auditor and approve users and track hackers after block the hackers in our application. Here the cloud admin will block the along with the details of the hacker.

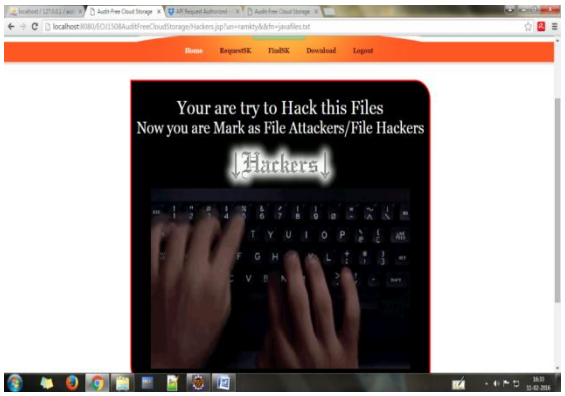


**Figure 4: Cloud Admin Tracking Hacker**

**Users:** Users register first, after login he must join any group. Each and Every Group Signature is Unique within the same group. User sign in the group login after download all files and not like this group he will revoke the group membership easily. Each and Every Users activity is tracked by Session Tracking Methodology. If the users try to hack the another users group signature and sign in the group login that time the user system configuration & the Users Details are intimated to cloud admin. Admin block the hack users after the user common username & password are going to invalid mode.

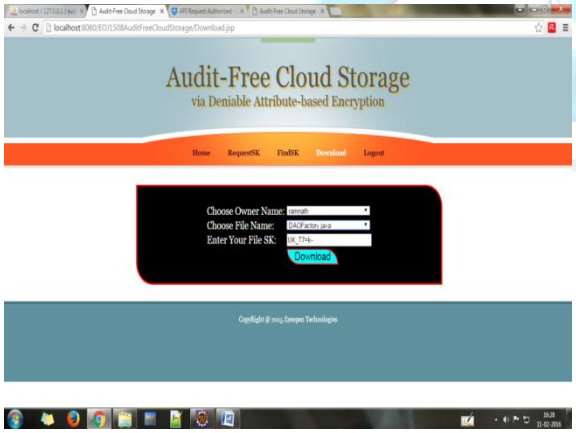


**Figure 5: User Providing Secret Key**

## VII. EXPERIMENTAL RESULTS

Accessing the cloud server via randomized key protocol is highly secured. In the randomized key algorithm, the unique key generated by the key distributor provides security to the users. This method uses an audit free storage, which skips the third party auditing of the data and files that has been uploaded onto the cloud server. The key generated varies for each file and for each user. This proposed protocol is thus highly secured compared to the existing system.

## VIII. CONCLUSIONS

In this proposed method, we provide an audit free cloud storage which is highly secured for the users accessing the cloud server using the randomized key protocol. This method makes coercion invalid. Our proposed scheme provides a possible way to fight against immoral interference with the right of privacy.

## IX. REFERENCES

[1] Po-Wen Chi, Chin-Laung Lei, "Audit-Free Cloud Storage via Deniable Attribute-based Encryption", IEEE Transactions on Cloud Computing, , no. 1, pp. 1, PrePrints PrePrints, doi:10.1109/TCC.2015.2424882

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Eurocrypt, 2005, pp. 457–473.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in ACM Conference on Computer and Communications Security, 2006, pp. 89–98.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in IEEE Symposium on Security and Privacy, 2007, pp. 321–334.

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

[6] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in Crypto, 2012, pp. 199–217.

[7] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in Public Key Cryptography, 2013, pp. 162–179.

[8] P. K. Tysowski and M. A. Hasan, "Hybrid attribute- and reencryption-based key management for secure and scalable mobile applications in clouds." IEEE T. Cloud Computing, pp. 172–186, 2013.

[9] Wired. (2014) Spam suspect uses google docs; fbi happy. [Online]. Available: http://www.wired.com/2010/04/cloud-warrant/ [9] Wikipedia. (2014) Global surveillance disclosures (2013present). [Online]. Available: http://en.wikipedia.org/wiki/Global surveillance disclosures (2013-present)

[10] (2014) Edward snowden. [Online]. Available: http://en. wikipedia.org/wiki/Edward Snowden

[11] (2014) Lavabit. [Online]. Available: http://en.wikipedia. org/wiki/Lavabit