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 III, March 2018- Available at www.ijraset.com

Framework for Secure Data Sharing in Dynamic Group Using Public Cloud

B. Manish Kumar[1], R. Raja Sekar[2], Mrs. D. Deepa[3], Mrs. P. Veeralakshmi[4]
[1, 2] *Department of Information Technology, Prince Shri Venkateshwara Padmavathy Engineering College, Chennai, India.*
[3]*Assistant Professor, Department of Computer Science and Engineering, Prince Shri VenkateshwaraPadmavathy Engineering College, Chennai, India.*
[4]*Associate Professor, Department of Information Technology, , Prince Shri Venkateshwara Padmavathy Engineering College, Chennai, India.*

Abstract: A public cloud is used to share the data to the users which makes the accessing operation much easier. A public cloud mainly supports the simultaneous data uploading/downloading. To provide security to the cloud, key agreement protocol have played a very important role in an efficient manner. Based on the proposed group data sharing model, we present general formulas for generating the random group key for multiple users.
Keywords: Key agreement Protocol, Random class, cloud.

I. INTRODUCTION

CLOUD computing and cloud storage have become hot topics in recent decades. Both are changing the way we live and greatly improving production efficiency in some areas.

At present, due to limited storage resources and the requirement for convenient access, we prefer to store all types of data in cloud servers, which is also a good option for companies and organizations to avoid the overhead of deploying and maintaining equipment when data are stored locally. The cloud server provides an open and convenient storage platform for individuals and organizations, but it also introduces security problems.

For instance, a cloud system may be subjected to attacks from both malicious users and cloud providers. In these scenarios, it is important to ensure the security of the stored data in the cloud. In this, several schemes were proposed to preserve the privacy of the outsourced data.

The above schemes only considered security problems of a single data owner. However, in some applications, multiple data owners would like to securely share their data in a group manner.

Therefore, a protocol that supports secure group data sharing under cloud computing is needed. A key agreement protocol is used to generate a common conference key for multiple participants to ensure the security of their later communications, and this protocol can be applied in cloud computing to support secure and efficient data sharing., a key agreement protocol is a protocol in which two or more parties can agree on a key in such a way that both Influence the outcome.

By employing the key agreement protocol, the conferees can securely send and receive messages from each other using the common conference key that they agree upon in advance.

Specifically, a secure key agreement protocol ensures that the adversary cannot obtain the generated key by implementing malicious attacks, such as eavesdropping.

Thus, the key agreement protocol can be widely used in interactive communication environments with high security requirements (e.g., remote board meetings, teleconferences, collaborative workspaces, radio frequency identification [5], cloud computing and so on).

A. Related Work

Existing cloud storage applications doesn't give complete data security. Replica of data is possible. Extra storage consumption resulting in the extra storage cost for data application in the cloud.

A secret key is needed to download the files from the cloud by using the group key. To decrypt the files, each user will get a individual private key. Encryption keys should be transmitted in a secure channel, which is not possible in practice, particularly in the open cloud environment. Cloud computing also leads to many security and privacy concerns, such as data integrity, confidentiality, reliability, fault tolerance.

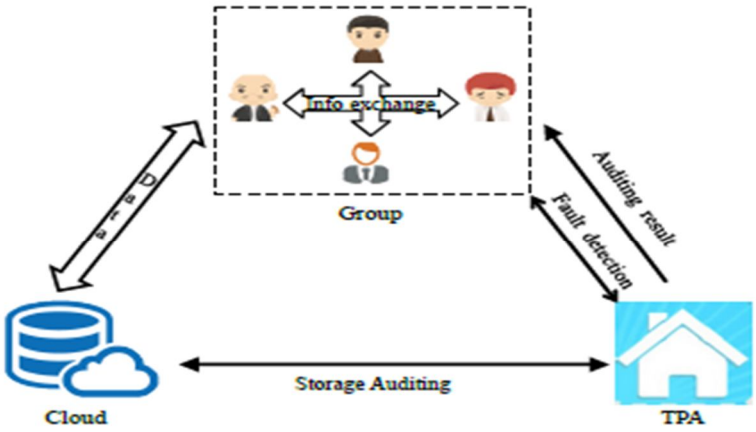


Fig. 1: System model of data sharing in cloud computing.

Fig 1.1 Architecture for Group Data Sharing

B. Proposed System

The users can securely obtain their private keys from group manager. User send request to group manager for access the wanted group, at that time our system provide individual secure key to user without activation. Then group manager see the requests and activate the keys after confirm them. After user's private key gets activation, then only user can access the group. Our scheme have fine-grained access control, any user in the group can use the source in the cloud and revoked users cannot access the cloud again after they are revoked. In our proposed system the group manager performs the below tasks when an new user joins the group or a user has left the particular group, Update the whole user name list. Generate a secure key and encrypt the key without activation and send to the updated user list. Update the rights in the cloud server.We proposed public cloud named **CloudMe** for data storage. Group manager makes sure that the revoked users cannot access the file if they conspire with untrusted cloud. The group manager creates the new encryption key for the specific group and transmits in an encrypted format using **key agreement protocol.**The System Architecture is given in Fig 3.1.

C. Des Algorithm

DES is a block cipher--meaning it operates on plaintext blocks of a given size (64-bits) and returns ciphertext blocks of the same size. Thus DES results in a permutation among the 2^64 (read this as: "2 to the 64th power") possible arrangements of 64 bits, each of which may be either 0 or 1.The representation of DES Permutation levels is given in the Fig 3.2. Each block of 64 bits is divided into two blocks of 32 bits each, a left half block L and a right half R. (This division is only used in certain operations.).The Structure of DES permutation is given in Fig 2.1.

The following is the pseudo code for DES Encryption,

```
function DES_Encrypt (M, K)    where M = (L, R)
   M ← IP(M)
   for round ←1 to 16 do
      Ki ← SK(K, round)
      L  ← L xor F(R, Ki)
      swap(L, R)
   end
swap(L, R)
   M ← IP-1(M)
   return M
end
```

The algorithm for decryption is similar. The only difference is the order in which the keys are used.

```
function DES_Decrypt (C, K)    where C = (L, R)
   C ← IP(C)

   for round ← 16 to 1 do
      Ki ← SK(K, round)
      L  ← L xor F(R, Ki)
      swap(L, R)
   end

swap(L, R)

   C ← IP-1(C)
   return C
end
```

Steps**:**

1) Fractioning of the text into 64-bit (8 octet) blocks.
2) Initial permutation of blocks.
3) Breakdown of the blocks into two parts: left and right, named *L* and *R*;
4) Permutation and substitution steps repeated 16 times (called **rounds**);
5) Re-joining of the left and right parts then inverse initial permutation.

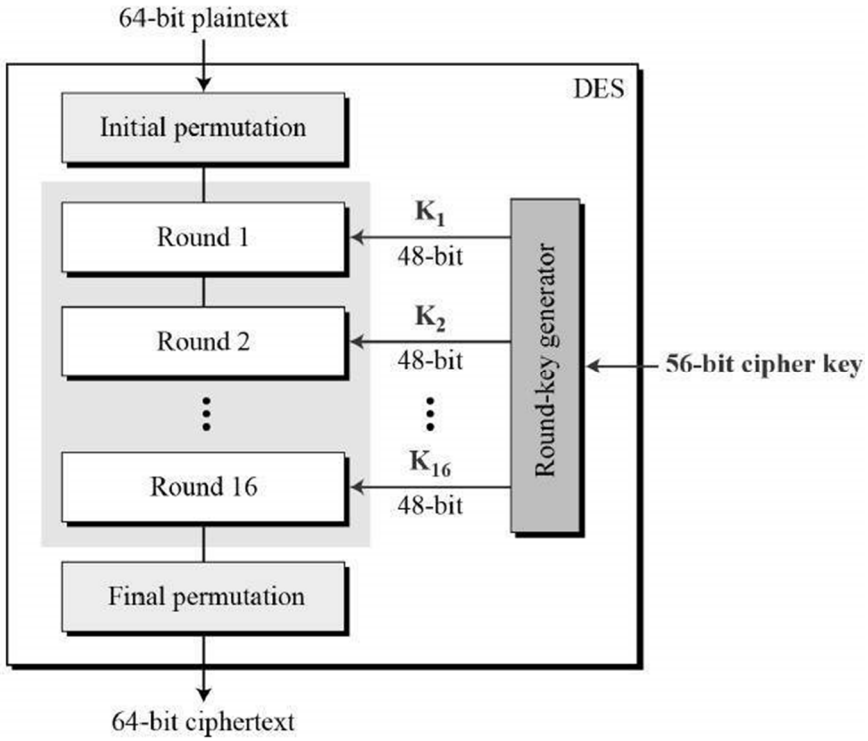


Fig 2.1 Structure of DES Permutation

D. Key Agreement Protocol

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. If properly done, this precludes undesired third parties from forcing a key choice on the agreeing parties. Protocols that are useful in practice also do not reveal to any eavesdropping party what key has been agreed upon. Many key exchange systems have one party generate the key, and simply send that key to the other party -- the other party has no influence on the key. Using a key-agreement protocol avoids some of the key distribution problems associated with such systems.

E. System architecture for proposed model:

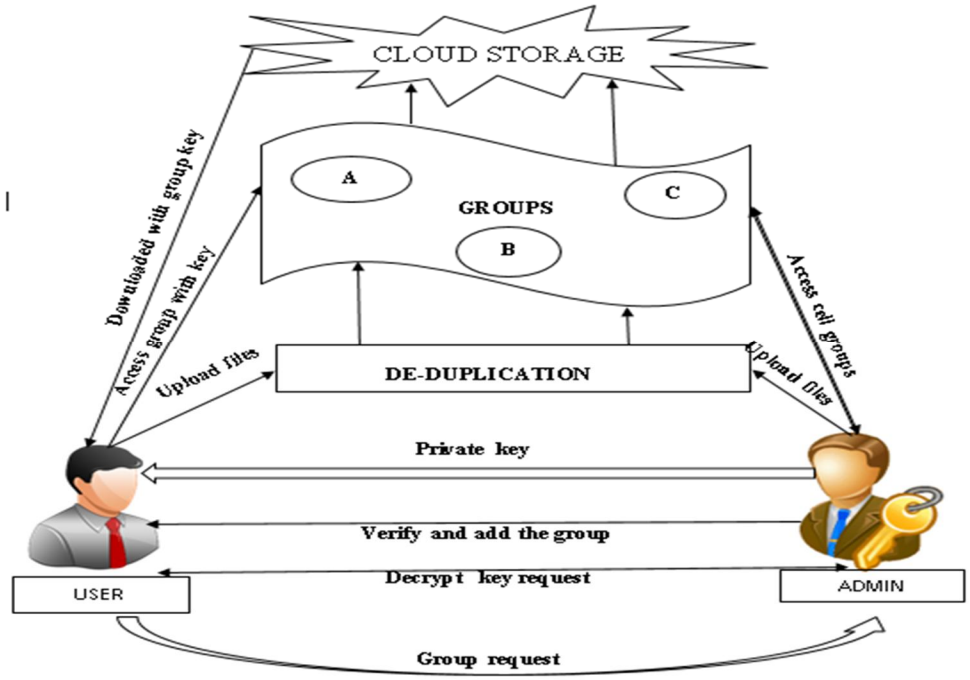


Fig 3.1 System Architecture for Secured Data Sharing using Public Cloud

F. Authentication

Anonymous key exchange, like Diffie–Hellman, does not provide authentication of the parties, and is thus vulnerable to man-in-the-middle attack. A wide variety of cryptographic authentication schemes and protocols have been developed to provide authenticated key agreement to prevent man-in-the-middle and related attacks. These methods generally mathematically bind the agreed key to other agreed-upon data, such as the following:

1) Public/private key pairs
2) Shared secret keys
3) Passwords

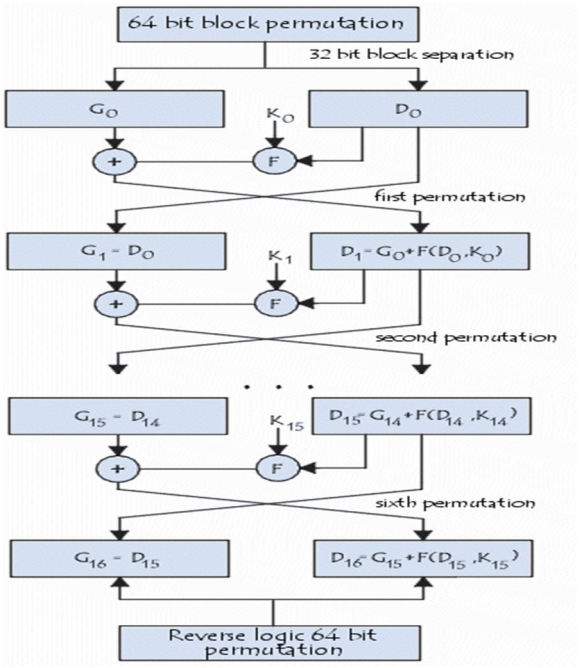


Fig3.2 Representation of DESPermutation

G. Fault Detection Phase

In practice, we cannot guarantee that all participants in the group are honest. The existence of malicious participants can seriously destroy the conference. In Yi's protocol, an attack from malicious participants is called a different key attack. In different key attacks, a malicious participant chooses different sub keys, generates different signatures and broadcasts different messages to different participants such that the signatures of malicious participants are valid and malicious participants can be authenticated by other participants. In addition, the different sub keys make different participants derive different conference keys, which may lead to serious damage of the conference and make the protocol invalid. Therefore, the fault detection phase is added to prevent different key attacks from malicious participants.

H. Shared Secret Keys

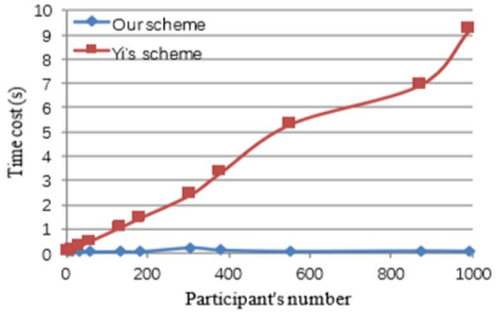
Secret-key cryptography requires the initial exchange of a shared key in a manner that is private and integrity-assured. When done right, man-in-the-middle attack is prevented. However, without the use of public-key cryptography, one may be left with undesirable key-management problems.

I. Passwords

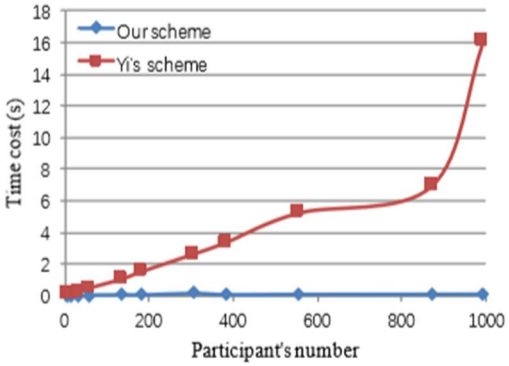
Password authenticated key agreement protocols require the separate establishment of a password (which may be smaller than a key) in a manner that is both private and integrity-assured. These are designed to resist man-in-the-middle and other active attacks on the password and the established keys.

J. Performance Evaluation

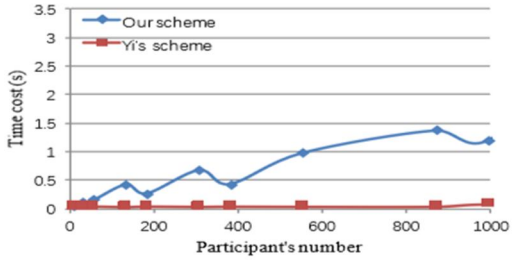
To study the performance of our scheme, we provide an experimental evaluation of the proposed scheme†. Our experiments are simulated by using C programming language with the pairing-based cryptography (PBC) library and the GUN multiple precision arithmetic (GMP) library on a VMware Workstation machine with Intel Core i5-3210 processors running at 2.50 GHz and 2 G memory.



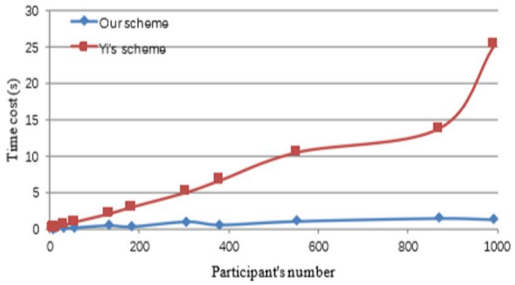
(a) Initial phase



(b) Key agreement phase



(c) Authentication phase



II. CONCLUSION

Thus the key agreement protocol that supports a Framework For Secure Data Sharing In Dynamic Group Using Public Cloud system has been utilized.By using this any user in particular group can do the activities like upload/download a file in a secured manner. In our future work, we would like to extend our protocol to provide more properties (e.g., anonymous users, traceability, and so on) to make it applicable for a variety of environments.

III. ACKNOWLEDGMENTS

We would like to thank our internal project guide Ms. D. Deepa, Assistant Professor of Department of Computer Science and Engineering and Ms. P. Veeralakshmi, Associate Professor, Department of Information Technology their guidance and suggestions during this work.

REFERENCES

[1] Jian Shen Tianqi Zhou, Debiao He, Yuexin Zhang, Xingming Sun and Yang Xiang, "Block Design-based Key Agreement for Group Data Sharing in Cloud Computing", IEEE transaction on Dependable and Secure Computing, vol. pp, no: 99, 12 July 2017.
[2] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," IEEE Transactions on Information Forensics and Security, vol. 10, no. 6, pp. 1–1, 2015.
[3] F. Chen, T. Xiang, Y. Yang, and S. S. M. Chow, "Secure cloud storage meets with secure network coding," in IEEE INFOCOM, 2014, pp. 673–681.
[4] Lan Zhou, Vijay Varadharajan, and Michael Hitchens,"Achieving Secure Role-Based Access Controlon Encrypted Data in Cloud Storage", inIEEE TRANSACTIONS VOL. 8, NO. 12, DECEMBER 2013.
[5] J. Shen, S. Moh, and I. Chung, "Identity-based key agreement protocol employing a symmetric balanced incomplete block design," Journal of Communications and Networks, vol. 14, no. 6, pp. 682–691, 2012.
[6] J. Yu, K. Ren, C. Wang, and V. Varadharajan,"Enabling cloudstorage auditing with key-exposure resistance," IEEE Transactionson Information Forensics and Security, vol. 10, no. 6, pp. 1–1, 2015.