ISSN:0975-9646

Efficient and Secure Scheme for Distributed Data Storage Systems

Y.Vijaya Ratna Kumari[1]
Department of Computer Science
M.V.R College of Engineering &Technology

T. Bindu Madhavi[2]
Assistant Professor
Department of Computer Science
M.V.R College Of Engineering & Tech

L.Ravi Kumar[3]
Assistant Professor
Department of IT
P.V.P Siddhartha Institute of Tech

Abstract: **The burden of maintaining a large number of files from the owner to proxy servers can be shifted by usage of secure distributed data storage. Generally proxy servers can convert encrypted files of the owner to encrypted files for the receiver without any knowledge on the content of the original files. It's a practice to remove the original files by the owner for the sake of space efficiency. Thus, the issues on confidentiality and integrity of the outsourced data must be addressed carefully. So, we propose two identity-based secure distributed data storage (IBSDDS) schemes. These schemes can capture the following properties: (1) The file owner can decide the access permission independently without the help of the private key generator (PKG); (2) A receiver can only access one file, instead of all files of the owner for one query; (3) Our schemes are secure against the collusion attacks, i.e., even if the receiver can compromise the proxy servers, intruder cannot obtain the owner's secret key. Though the first scheme is only secure against the chosen plaintext attacks (CPA), the second scheme is secure against the chosen cipher text attacks (CCA). To the best of our knowledge, it is the first IBSDDS schemes where an access permissions is made by the owner for an exact file and collusion attacks can be protected in the standard model. Experimental results validating our approach are also presented.**

Keywords: **Distributed Data Storage, Identity-based System, Access Control, Security**

INTRODUCTION

Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. If a cloud user accesses services on the infrastructure layer, for example, one can run her own applications on the resources of a cloud infrastructure and remain responsible for the maintenance, support and security of these applications him/herself. If one accesses a service on the application layer, these are normally taken care of by the cloud service provider. Some of the benefits of cloud computing are: Reduce spending on technology infrastructure, Achieve economies of scale, Streamline processes, Reduce capital costs, Globalize your workforce on the cheap, Improve accessibility, Monitor projects more effectively, Minimize licensing new software, Improve flexibility. Users are especially concerned on the integrity, confidentiality, and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems because the cloud is managed by an untrusted third party. Therefore, the challenge here is to guarantee that outsourced files are not accessed by the unauthorized users and not modified by proxy in the data storage research community. Moreover, how to guarantee that an authorized user can query the outsourced files from proxy servers is another concern as the proxy server only maintains the outsourced cipher texts. Research around these topics grows significantly.

LITERATURE REVIEWS

A. Executing SQL over encrypted data in the database-service-provider model [1]

Rapid advances in networking and Internet technologies have fueled the emergence of the "software as a service" model for enterprise computing. Examples of commercially viable software services include electronic mail services, general storage services, rent-a-spreadsheet, disaster protection services. Using the "Database as a Service" model users can create, store, modify, and retrieve data from anywhere, if they have access to the Internet. This introduces several challenges, an important issue being data privacy. In this context we specifically address the issue of data privacy. The two main privacy issues are: 1) The owner of the data needs to be assured that the data stored on the service-provider site is protected against data thefts from outsiders. 2) Since the providers themselves cannot be trusted, data needs to be protected even from the service providers. In this paper, we focus on the second challenge. We explore techniques to execute SQL queries over encrypted data. Our strategy here is to process as much of the query as possible at the service providers' site, without having need to decrypt the data. The remainder of the query processing and decryption are performed at the client site. We explore an algebraic framework to split the query to minimize the computation at the client site. Experimental results validating our approach are also presented.

B. Chip-secured data access: Confidential data on untrusted servers [2]

The democratization of ubiquitous computing (access data anyhow, anywhere, anytime), the increasing connection of corporate databases to the Internet and also now a day's natural companies of resort to Web hosting emphasizes the need for data restricted. Database servers arouse user's suspicion because no one can fully trust traditional security mechanisms against more and more frequent and malicious attacks and no one will be expert on administering an invisible D B A data confidential. This paper gives a deep analysis of existing security solutions and concludes on the intrinsic weakness of the traditional server-based approach to preserve data confidentiality. With this idea in mind, we introduce a solution that enforces data confidentiality and controls personal

privileges of thanks to a client based security component which acts as a mediator between a client and an encrypted database. This solution is called C-SDA(Chip-Secured Data Access). This component is inserted in a plastic card with integrated circuits to prevent any tampering to occur. This combination of hardware and software security components constitutes a strong guarantee against attacks threatening personal as well as business data.

C. How to build a trusted database system on untrusted storage [3]

Some coming into existence applications need programs to maintain sensitive state on untreated hosts. This paper presents the design and development of a trusted database system. T D B, which occupies a small amount of trusted space to protect an upgraded amount of untreated space. The database is encrypted and validated against a collision-resistant hash kept in trusted space, so untreated programs cannot read the database or change it. T D B integrates encryption and hashing with a bottom-level data model, which protects data and metadata same, dislike systems built on top of a conventional database system. The development exploits synergies between hashing and log-structured storage. Initial performance results show that T D B outperforms an off-the-shelf embedded database system, thus supporting the suitability of the T D B design.

D. Improved proxy re-encryption schemes with applications to secure distributed storage [4]

In 1998, Blaze, Bloomer, and Strauss (BBS) proposed an application called atomic proxy re-encryption, in which a semi trusted entity converts a cipher text for Alice into a cipher text for Bob without seeing the underlying plain text. We predict that fast and secure re-encryption will become increasingly popular as a method for managing encrypted file systems. Although efficiently determine, the wide-spread adoption of BBS re-encryption has been hindered by considerable security risks. Following recent work of Dodoes and Ivan, we present new re-encryption schemes that realize a stronger notion of security and demonstrate the usefulness of proxy re-encryption as a method of adding access control to a secure file system. Performance levels of our experimental file system demonstrate that proxy re-encryption can work effectively in practice.

E. Effficient and private access to outsourced data [5]

There is an increasing need for novel techniques that support not only data confidentiality, also confidentiality of the accesses that users make on such data as the use of external storage and data processing services for storing and managing sensitive data is becoming more and more common. In this paper, we propose a technique for access, guaranteeing content and pattern confidentiality in the data outsourcing scenario. The proposed technique adapts traditional B+-tree which introduces a shuffle index structure. We show that our solution exhibits a limited performance cost resulting effectively usable in practice.

EXISTING SYSTEM

Cloud computing provides users with a convenient mechanism to manage their personal files with the notion called database-as-a-service (DAS). In DAS schemes, a user can outsource his encrypted files to untrusted proxy servers. Proxy servers perform functions on the outsourced cipher texts without any knowledge about the original files. However, technique has not been employed extensively. Main reason is that users are especially concerned on the integrity, confidentiality and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, since the cloud is managed by an untrusted third party. Once the files are outsourced to proxy servers, the user will remove the files from his local machine. The challenge here is, how to guarantee the outsourced files are not accessed by the unauthorized users and not modified by proxy servers. This is an important problem that has been considered in the data storage research community. Further, how to guarantee that an authorized user can query the outsourced files from proxy servers is another concern as the proxy server only maintains the outsourced cipher texts. Research around these topics grows significantly.

Problem Statement

Users are especially concerned on the integrity, confidentiality and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems because the cloud is managed by an untrusted third party. An important problem that has been considered in the data storage research community is outsourced files are not accessed by the unauthorized users and not modified by proxy servers.

PROPOSED SYSTEM

We propose two identity-based secure distributed data storage (IBSDDS) schemes in this paper. Standard model is for each query the receiver can only access one of the owner's files, instead of all files. That is, access permission (re-encryption key) is bound not only to the identity of the receiver but also the file i.e. it can be decided by the owner, not by trusted party (PKG).Moreover, our schemes are secure against the collusion attacks. The scheme is advantageous because:

- It has two schemes of security, the first scheme is CPA secure and second one achieves CCA security.
- We believe that this is the first IBSDDS schemes where access permission is made by the owner for an exact file and collusion attacks can be protected in the standard model.
- To achieve a stronger security and implement file based access control, owner must be online to authenticate requesters and also to generate access permissions for them. Hence, the owner in our schemes needs do more computations than that in PRE schemes. However PRE schemes can provide the similar functionalities of our schemes when the owner only has one file, these are not practical and flexible.

SYSTEM ARCHITECTURE AND DESIGN

In this section, we propose an identity-based secure distributed data storage (IBSDDS I) scheme which is secure against chosen plaintext attacks (CPA). At first, the file owner encrypts his files and outsources the cipher texts

to the proxy servers. The proxy servers validate the outsourced cipher texts and store them for the owner.

For one query, the receiver computes authentication information (AI) using his secret key and sends it to the proxy server. Proxy server sends the identity of the receiver, AI and the partial intended cipher text to the owner. Suppose that the owner can know which file the receiver wants to access from the partial cipher text. To check whether the requester is a legal user in the system, the owner validates the the received AI. If the AI is correct, the owner computes an access permission (re-encryption key) using his secret key, the partial cipher text and the identity of the receiver, and sends it to the proxy server. Else, the access is denied. The proxy sever transfers the intended cipher text to a cipher text for the receiver using the received access permission. Finally, the receiver can decrypt the re-encrypted cipher text by his secret key and obtains the original file. Fig.1 explains the model of our IBSDDS schemes.

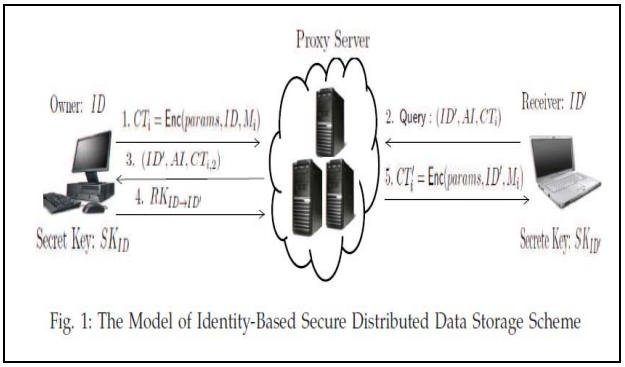

Figure 1: IBSDDS scheme

There are four items in an identity-based secure distributed data storage (IBSDDS) scheme: the private key generator (PKG), the data owner, the proxy server and the receiver. PKG validates the users' identities and issues secret keys to them. The data owner encrypts his data and outsources the cipher texts to the proxy servers. Figure 2 shows the input design of the same

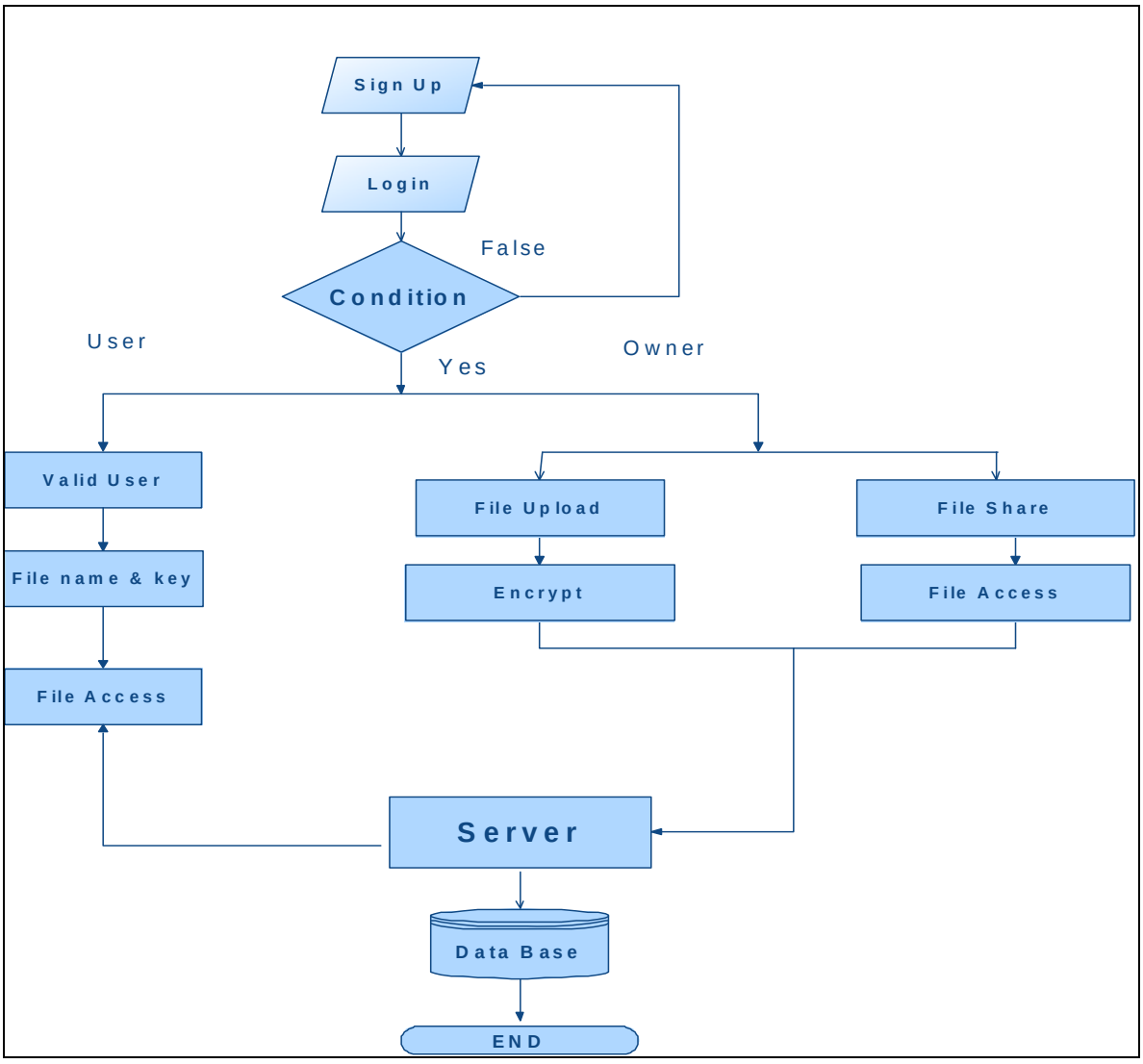

Figure 2: The Data Flow Diagram

EXPERIMENTAL RESULTS

Input design focuses on controlling the amount of input required, errors, avoiding delay, avoiding extra steps and keeping the process simple. In any system results of processing are communicated to the users and to other system through outputs which is the most important and direct source information to the user. Intelligent and efficient output design improves the system's relationship to help user decision-making. Below points show the results using IBSDDS scheme.

1. Input is designed in such a way so that it provides security and ease of use with retaining the privacy.
2. The data entry screen as in Figure 3 is designed in such a way that all the data manipulates can be performed. Records viewing facilities also is provided.

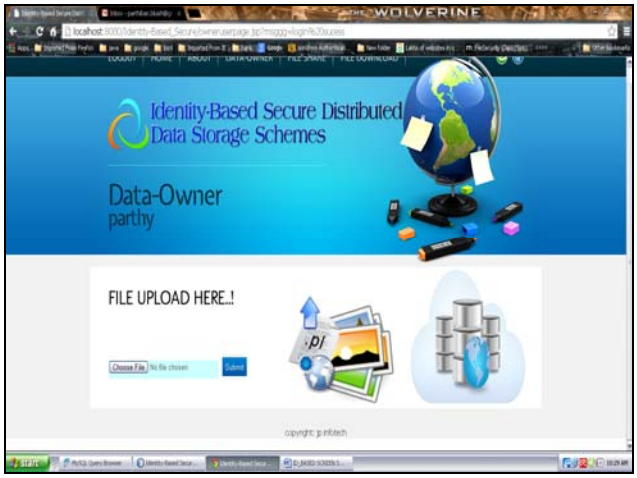

Figure 3: File Upload

3. When file is uploaded we will check for its validity and appropriate messages are provided as when needed to the user. Thus the objective of input design is to create an input layout that is easy to follow as in Figure 4.

Figure 4: File Share

4. The uploaded file is encrypted using the IDBSS scheme and the encrypted format as in Figure 5.

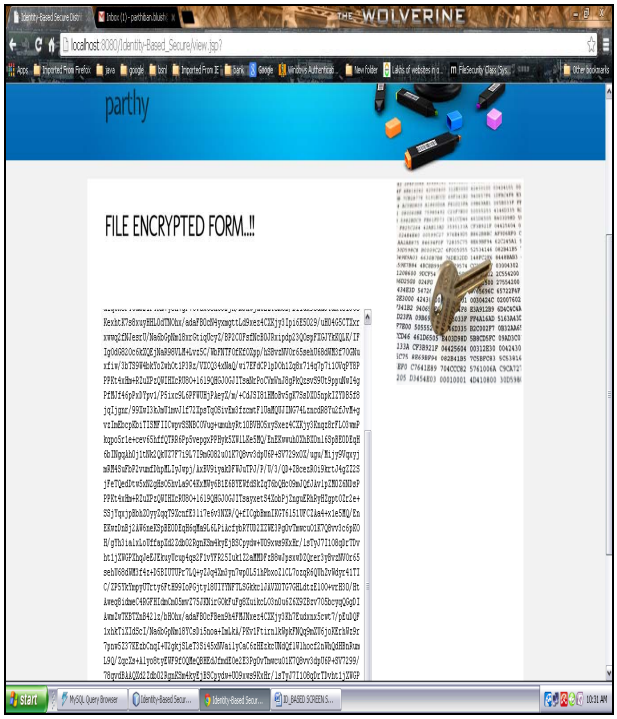

Figure 5: Encrypted file using the IBSDDS scheme

5. The Data user of can download the file using the secret key. The secret key is validated using IBSDDS scheme and the user is granted access only if the key is valid Figure 6 & 7.

Figure 6: User with correct secret key

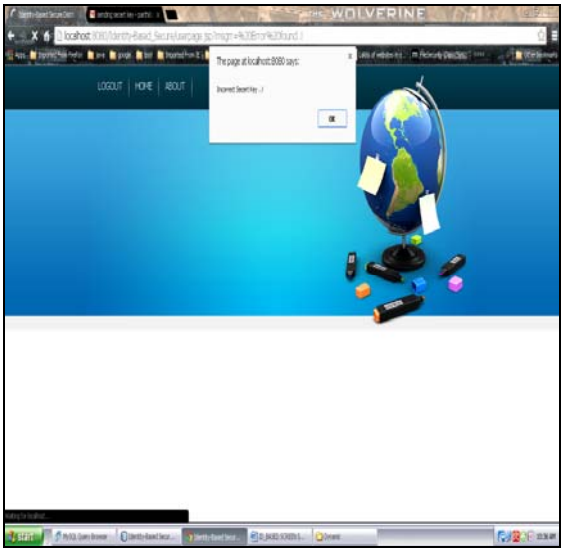

Figure 7: User with wrong secret key

CONCLUSION

Distributed data storage schemes provide the users with convenience to outsource their files to untrusted proxy servers in efficient and secured manner. Identity-based secure distributed data storage (IBSDDS) schemes are a special kind of distributed data storage schemes where users are identified by their identities and can communicate without the need of verifying the public keys. In this paper, we proposed two new IBSDDS schemes in standard model where, for each query, the receiver can only access one file, instead of all files. Further, the access permission can be made by the owner, instead of the trusted party. Remarkably, our schemes are secure against the collusion attacks. First scheme is CPA secure, while the second one is CCA secure.

REFFERENCES

[1] H. Hacig¨um¨us, B. R. Iyer, C. Li, and S. Mehrotra, “Executing SQL over encrypted data in the database-service-provider model,” in Proceedings: SIGMOD Conference - SIGMOD’02, June 2002, vol. 2002, pp. 216–227.

[2] L. Bouganim and P. Pucheral, “Chip-secured data access: Confidential data on untrusted servers,” in *Proc. International Conference on Very Large Data Bases - VLDB’02*, Aug 2002, pp. 131– 142.

[3] U. Maheshwari, R. Vingralek, and W. Shapiro, “How to build a trusted database system on untrusted storage,” in *Proc. Symposium on Operating System Design and Implementation - OSDI’00*, Oct. 2000, pp. 135–150.

[4] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” in *Proc. Network and Distributed System Security Symposium - NDSS’05*, Feb 2005, pp. 1–15.

[5] S. D. C. di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, and P. Samarati, “Effficient and private access to outsourced data,” in *Proc. International Conference on Distributed Computing Systems - ICDCS’11*, IEEE, Jun. 2011, pp. 710–719.