IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. IV (Nov – Dec. 2014), PP 53-59
www.iosrjournals.org

A Novel identity based secure distributed data storage scheme

M. Ramadevi[1], D. Bulla Rao[2]

[1]*M.Tech Student Computer Science Engineering, SITS, JNTU-A, Tirupathi, AP*
[2]*Assistant Professor, Dept. of CSE, SITS, JNTU-A, Tirupathi, AP*

Abstract: *Secure distributed data storage can shift the burden of maintaining a large number of files from the owner to proxy servers. Proxy servers can convert encrypted files for the owner to encrypted files for the receiver without the necessity of knowing the content of the original files. In practice, the original files will be removed by the owner for the sake of space efficiency. Hence, the issues on confidentiality and integrity of the outsourced data must be addressed carefully. In this paper, we propose two identity-based secure distributed data storage (IBSDDS) schemes. Our 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) For one query, a receiver can only access one file, instead of all files of the owner; (3) Our schemes are secure against the collusion attacks, namely even if the receiver can compromise the proxy servers, he cannot obtain the owner's secret key. Although 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 access permission is made by the owner for an exact file and collusion attacks can be protected in the standard model.*

Keywords: *Distributed data, identity, key generator, cipher text attacks*

I. Introduction

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 can perform some functions on the outsourced ciphertexts without knowing anything about the original files. Unfortunately, this technique has not been employed extensively. The main reason lies in that users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party.

After outsorcing the files to proxy servers, the user will remove them from his local machine. Therefore, how to guarantee the outsoured files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community. Furthermore, 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 ciphertexts. Consequently, research around these topics grows significantly.

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

II. Cloud Computing

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers. The structure of cloud computing was shown in figure 1.

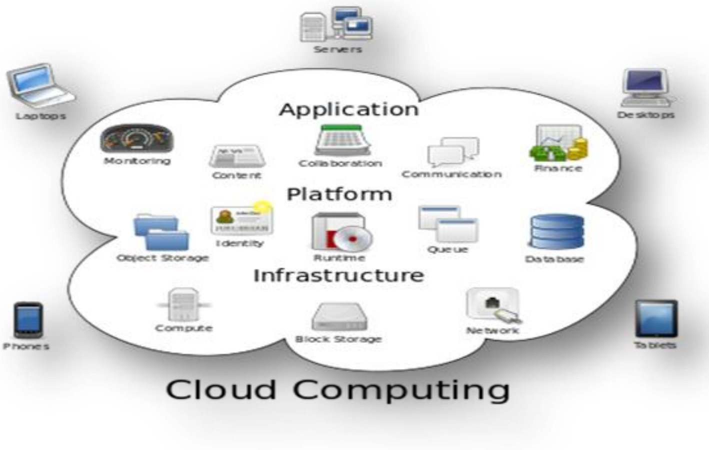

Fig. 1 Structure of cloud computing

A. How Cloud Computing Works?

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

B. Characteristics and Services Models

The figure 2 shows the characteristics of cloud computing. The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

- On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.
- Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

On-demand self-service
Ubiquitous network access
Location transparent resource pooling
Rapid elasticity
Measured service with pay per use

Fig. 2 Characteristics of cloud computing

C. Characteristics of cloud computing

Services Models:

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. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.

D. Benefits of cloud computing:

1. Achieve economies of scale – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.
2. Reduce spending on technology infrastructure. Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
3. Globalize your workforce on the cheap. People worldwide can access the cloud, provided they have an Internet connection.
4. Streamline processes. Get more work done in less time with less people.
5. Reduce capital costs. There's no need to spend big money on hardware, software or licensing fees.
6. Improve accessibility. You have access anytime, anywhere, making your life so much easier!
7. Monitor projects more effectively. Stay within budget and ahead of completion cycle times.
8. Less personnel training is needed. It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.
9. Minimize licensing new software. Stretch and grow without the need to buy expensive software licenses or programs.

III. Proposed Framework

In this paper, we propose two identity-based secure distributed data storage (IBSDDS) schemes in standard model where, for one query, the receiver can only access one of the owner's files, instead of all files. In other words, access permission (re-encryption key) is bound not only to the identity of the receiver but also the file. The access permission can be decided by the owner, instead of the trusted party (PKG). Furthermore, our schemes are secure against the collusion attacks.

It has two schemes of security, the first scheme is CPA secure, and the second scheme achieves CCA security. To the best of our knowledge, it 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, the owner must be online to authenticate requesters and also to generate access permissions for them. Therefore, the owner in our schemes needs do more computations than that in PRE schemes. Although PRE schemes can provide the similar functionalities of our schemes when the owner only has one file, these are not flexible and practical.

A total of four modules are present in the proposed framework i.e., Data Owner, Private key Generator, Proxy Server and The Receiver Module. In data owner module, first the new data owner registers and then gets a valid login credentials. After logged in, the data owner has the permission to upload their file into the Cloud Server. The data owner encrypts his data and outsources the ciphertexts to the proxy servers. In Private Key Generator module, the private key generator (PKG) validates the users' identities and issues secret keys to them. The key is generated and sent to their respective mail ids with the file name and the corresponding key values.

Proxy servers store the encrypted data and transfer the cipher text for the owner to the cipher text for the receiver when they obtain access permission (re-encryption key) from the owner. In these systems, proxy servers are assumed to be trusted. They authenticate receivers and validate access permissions. The interactions between the proxy servers and receivers are executed in a secure channel. Therefore, these systems cannot provide an end-to-end data security, namely they cannot ensure the confidentiality of the data stored at the proxy server. In these schemes, a receiver authenticates himself to the proxy server using his password. Then, the proxy server passes the authentication result to the file owner. The owner will make access permission according to the received information.

The receiver authenticates himself to the owner and decrypts the re-encrypted Ciphertext to obtain the data. In these systems, an end to-end security is provided by cryptographic protocols which are executed by the file owner to prevent proxy servers and unauthorized users from modifying and accessing the sensitive files. These systems can be divided into two types: shared file system and non-shared system. In shared file systems the owner can share his files with a group of users. Cryptographic techniques deployed in these systems are key sharing, key agreement and key revocation. In non-shared file systems in order to share a file with another user, the owner can compute an access key for the user using his secret key. In these two systems, the integrity of the sensitive files is provided by digital signature schemes and message authentication codes (MAC).

The data flow diagram, Use case diagram, and activity diagram are shown in figures from 3 to 5 respectively.

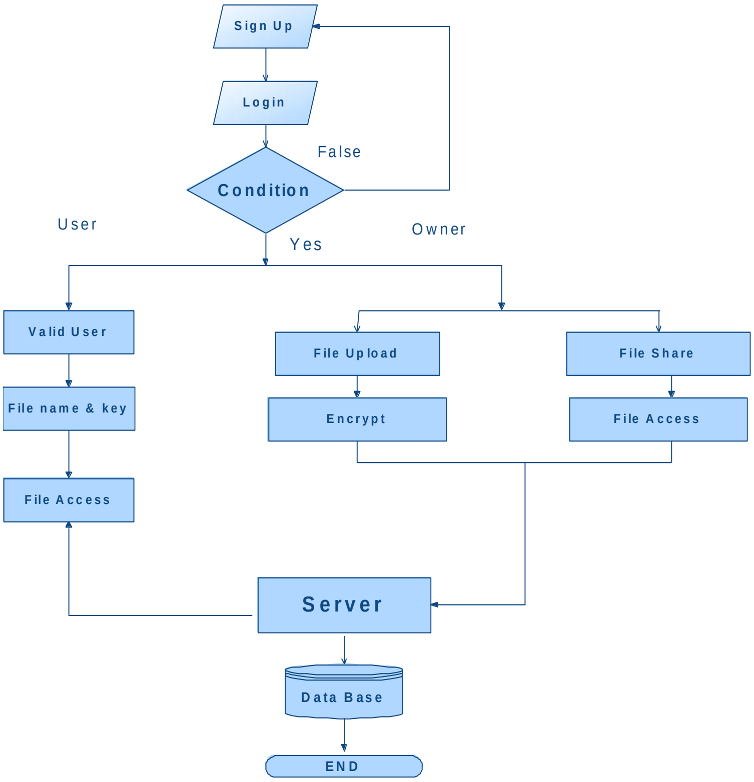

Fig. 3 Data flow diagram

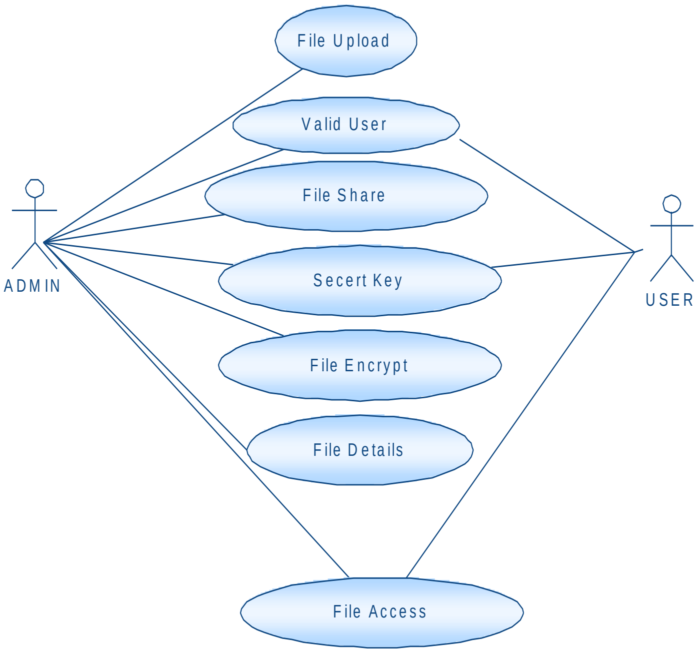

Fig. 4 Use case diagram

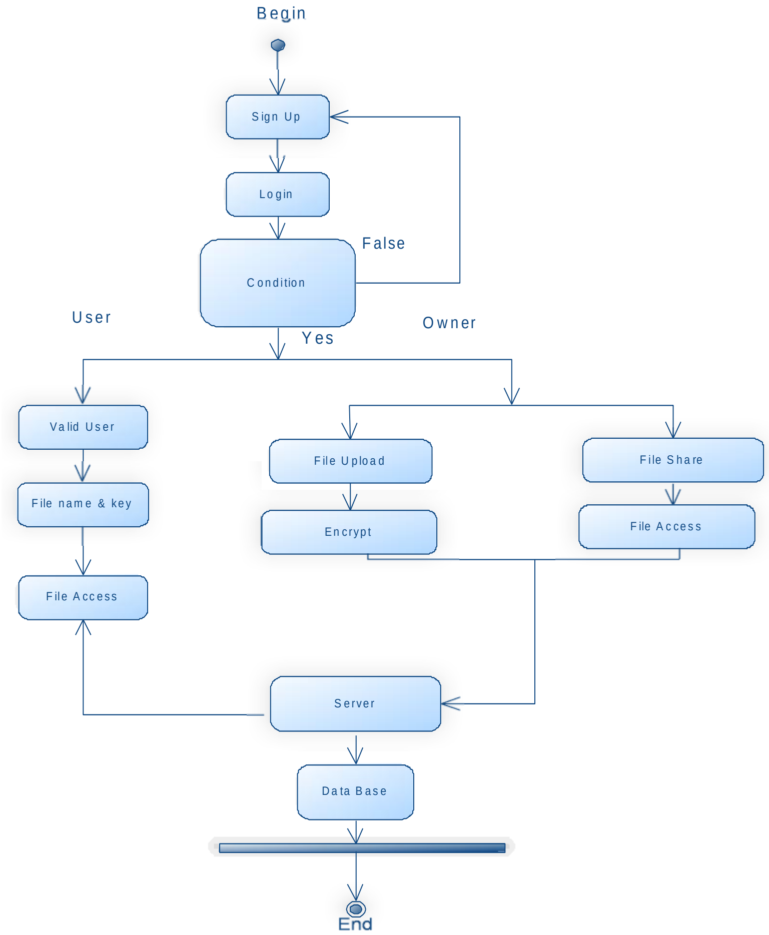

Fig. 5 Activity diagram

IV. Io Design

A. Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- ➢ What data should be given as input?
- ➢ How the data should be arranged or coded?
- ➢ The dialog to guide the operating personnel in providing input.
- ➢ Methods for preparing input validations and steps to follow when error occur.

Objectives

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

B. Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

V. Conclusions

Distributed data storage schemes provide the users with convenience to outsource their files to untrusted proxy servers. 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 key certificates. In this paper, we proposed two new IBSDDS schemes in standard model where, for one query, the receiver can only access one file, instead of all files. Furthermore, the access permission can be made by the owner, instead of the trusted party. Notably, our schemes are secure against the collusion attacks. The first scheme is CPA secure, while the second one is CCA secure

References

[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 (M. J. Franklin, B. Moon, and A. Ailamaki, eds.), vol. 2002, (Madison, Wisconsin, USA), pp. 216–227, ACM, Jun. 2002.

[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, (Hong Kong, China), pp. 131– 142, Morgan Kaufmann, Aug. 2002.

[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, (San Diego, California, USA), pp. 135–150, USENIX, Oct. 2000.

[4] A. Ivan and Y. Dodis, "Proxy cryptography revisited," in Proc. Network and Distributed System Security Symposium - NDSS'03, (San Diego, California, USA), pp. 1–20, The Internet Society, Feb. 2003.

[5] 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, (San Diego, California, USA), pp. 1–15, The Internet Society, Feb. 2005.

[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Transactions on Information and System Security, vol. 9, no. 1, pp. 1–30, 2006.

[7] 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, (Minneapolis, Minnesota, USA), pp. 710–719, IEEE, Jun. 2011.

[8] H.-Y. Lin and W.-G. Tzeng, "A secure erasure code-based cloud storage system with secure data forwarding," IEEE Transactions on Parallel and Distributed Systems, Digital Object Indentifier 10.1109/TPDS.2011.252 2012.

[9] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. Advances in Cryptology - ASIACRYPT'08 (J. Pieprzyk, ed.), vol. 5350 of Lecture Notes in Computer Science, (Melbourne, Australia), pp. 90–107, Springer, Dec. 2008.

[10] A. Juels and B. S. K. Jr., "PORs: Proofs of retrievability for large files," in Proceedings: ACM Conference on Computer and Communications Security - CCS'07 (P. Ning, S. D. C. di Vimercati, and P. F. Syverson, eds.), (Alexandria, Virginia, USA), pp. 584–597, ACM, Oct. 2007.

[11] Y. Dodis1, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in Proc. Theory of Cryptography Conference - TCC'09 (O. Reingold, ed.), vol. 5444 of Lecture Notes in Computer Science, (San Francisco, CA, USA), pp. 109–127, Springer, Mar. 2009.

[12] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in Proc. ACM Cloud Computing Security Workshop - CCSW'09, (Chicago, Illinois, USA), pp. 43–53, ACM, Nov. 13 2009.

[13] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in Proc. Advances in Cryptology - ASIACRYPT'09 (M. Matsui, ed.), vol. 5912 of Lecture Notes in Computer Science, (Tokyo, Japan), pp. 319–333, Springer, Dec. 2009.

[14] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. ACM Conference on Computer and Communications Security - CCS'07 (P. Ning, S. D. C. di Vimercati, and P. F. Syverson, eds.), (Alexandria, Virginia, USA), pp. 598–610, ACM, Oct. 2007.

[15] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. International conference on Security and privacy in communication netowrks - SecureComm'08, (Istanbul, Turkey), Sep., ACM, 2008.

[16] C. C. Erway, A. K¨upc¨u, C. Papamanthou, and R. Tamassia, “Dynamic provable data possession,” in Proc. ACM Conference on Computer and Communications Security - CCS’09 (E. Al-Shaer, S. Jha, and A. D. Keromytis, eds.), (Chicago, Illinois, USA), pp. 213–222, ACM, Nov. 2009.

[17] B. Carbunar and R. Sion, “Toward private joins on outsourced data,” IEEE Transactions on Knowlege and Data Engineering, vol. 9, no. 24, pp. 1699–1710, 2012.

[18] J. Hur, “Improving security and efficiency in attribute-based data sharing,” IEEE Transactions on Knowlege and Data Engineering, p. Digital Object Indentifier 10.1109/TKDE.2011.78.

[19] J. Hur and D. K. Noh, “Attribute-based access control with efficient revocation in data outsourcing systems,” IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 7, pp. 1214–1221, 2011.