International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 6 ǁ 2022 ǁ PP. 624-629

A Higher-Level Security Scheme for Key Access on Cloud Computing

G. SAJID
K RAJASEKHAR REDDY,
PG SCHOLORS ,DR.M.SARAVANAMUTHU (ASSISTANT PROFFESSOR)
MASTER OF COMPUTER APPLICATION
MADANAPALLE INSTITUTE OF TECHNOLOGY&SCIENCE, ANDHRAPRADHESH 517291

Abstract: *In this work, we construct a key access management scheme that seamlessly transitions any hierarchical-like access policy to the digital medium. The proposed scheme allows any public cloud system to be used as a private cloud. We consider the data owner an entity consisting of several organization units. We provide a secure method for each user of this entity to access the public cloud from both inside and outside the company's network. The idea of our key access control scheme, which is based on Shamir's secret sharing algorithm and polynomial interpolation method, is suitable especially for hierarchical organizational structures. It offers a secure, flexible, and hierarchical key access mechanism for organizations utilizing mission-critical data. It also minimizes concerns about moving mission-critical data to the public cloud and ensures that only users with sufficient approvals from the same or higher privileged users can access the key by making use of the topological ordering of a directed graph, including self-loop. Main overheads such as public and private storage needs are reduced to a tolerable level, and the key derivation is computationally efficient. From a security perspective, our scheme is both resistant to collaboration attacks and provides key in distinguishability security. Since the key does not need to be held anywhere, the problem of a data breach based on key disclosure risk is also eliminated.*

Date of Submission: 29-05-2022 Date of acceptance: 10-06-2022

I. INTRODUCTION

Digitizing several services increase demands on storage systems, large-scale computations, and hosting. In addition, advances in networking technology and administrative difficulties lead companies to outsource these services. A relatively new method called cloud computing enables users to access services from any location at any time [1]. In this work, we design a novel scheme to access a cloud storage system that runs on third parties' cloud infrastructure. The proposed method provides a secure scheme so that organizations requiring a higher level of security can use any public cloud infrastructure. Cloud computing also includes various service models [2] such as infrastructure as a service (IaaS), where a customer consumes a provider's computing, storage, and network resources; platform as a service (PaaS) where a customer uses the provider's ready-made environments to develop, run, and manage specific applications; and software as a service (SaaS) where a customer runs software on the infrastructure of the providers. The work [3] adds a service model, which is called network as a service (NaaS), where the customers are provided transport connectivity and related network services. In addition, communication as a service (CaaS), compute as a service (CompaaS), data storage as a service (DSaaS) are defined in [4], and in this work, we focus on the DSaaS model. Cloud deployment models are categorized as private, public, community, and hybrid cloud [2]–[4]. The public cloud is defined to be a multi-tenant environment where the cloud computing environment is shared with several other users. The private cloud is a single-tenant environment where the hardware, storage, and network are dedicated to a single user. The community cloud is provided for private use by a specific consumer community and is owned, managed, and operated by the organizations in the community. In addition, the hybrid cloud is a composition of two or more distinct cloud deployment models. In a public cloud, generally, compliance, security, and privacy requirements can create an issue since the infrastructure. is managed and owned by a cloud storage provider that is located off-premise. The system can be accessed by any user who pays for the service. On the other hand, in the private cloud, these requirements do not generally create an issue since the infrastructure which is managed and owned by the customer is located on-premise. Many organizations are slowing down their overall public cloud adoption plans even though public cloud infrastructure ensures many advantages, especially in total cost .

II. SYSTEM ANALYSIS & FEASIBILITY STUDY

Existing Method:

In existing system, we are using public cloud .In a public cloud, generally, compliance, security, and privacy requirements can create an issue since the infrastructure is managed and owned by a cloud storage provider that is located off-premise. The system can be accessed by any user who pays for the service. On the other hand, in the private cloud, these requirements do not generally create an issue since the infrastructure which is managed and owned by the customer is located on-premise.

Disadvantages:

- Security issues
- Data privacy issues
- All users data can be store in same cloud location

Proposed System:

In proposed system we are using private cloud. The private cloud is a single-tenant environment where the hardware, storage, and network are dedicated to a single user. The community cloud is provided for private use by a specific consumer community and is owned, managed, and operated by the organizations in the community.

ADVANTAGES

- Increasing data privacy
- Increasing security
- Each user can have individual data storage

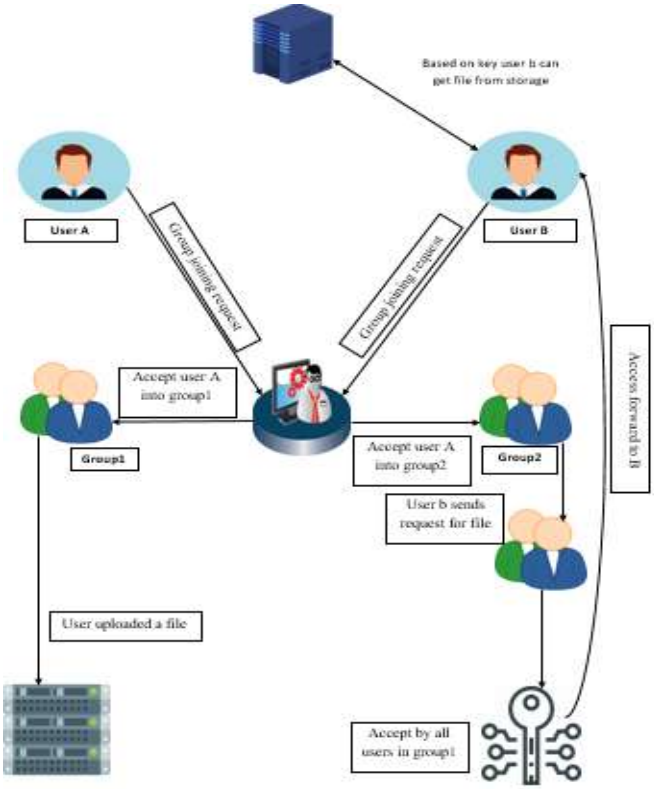

Fig : User Control Flow diagram

MODULES

In this project there is four modules

1. Cloud Service provider
2. Users

Cloud Service Provider:

Login: Login with valid email id and password

Add Group: CSP can add groups. In the group adding form contains group name

View Group: CSP can view all the groups which all added.

User Request: CSP can view user requests.

Logout: Finally CSP Logout
Users:
Register: User will register with details like name, email, password, address etc.
Login: User will login into the system using registered details.
View Groups: Views all the groups and add into the group
Upload Files: User can upload files in the group.
View Files: user can view all the files and send's request
Send Request: User can send to the file and directly downloads if the users are same group
Otherwise group members all has to accept the request then only file can be downloaded
Logout: Finally logout

III. METHODOLOGY AND ALGORITHMS:

Shamir's Secret Sharing Algorithm:

Shamir's Secret Sharing (SSS) is a key distribution algorithm. It is named for the well-known Israeli cryptographer Adi Shamir who co-invented the Rivest–Shamir–Adleman (RSA) algorithm.

SSS divides a secret, such as a crypto key, into parts called shares. The shares are distributed to a group of people who are parties to the conversation. The parts of the secret are brought together to reconstruct the secret, but an important feature of Shamir's Secret Sharing is that the total number of shares is not needed to reconstruct the secret. A number less than the total number, called the threshold, is required. This helps avoid failures in decrypting the closely-held information should just one or a few parties be unavailable.

SSS is practical in its solution to the key-sharing problems many arrangements face, and is therefore usually used to secure the keys to something that is encrypted or secure using other tools or algorithms. A simple illustration of SSS is that of a vault that only a corporate board may access. The passcode is encrypted by SSS, so a quorum (threshold) of board members is needed to authorize the display or release of the vault passcode. If a board member is traveling, but the threshold is met, SSS still allows for a reasonable assurance that the vault is secure.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.
There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.
Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

IV. Result:

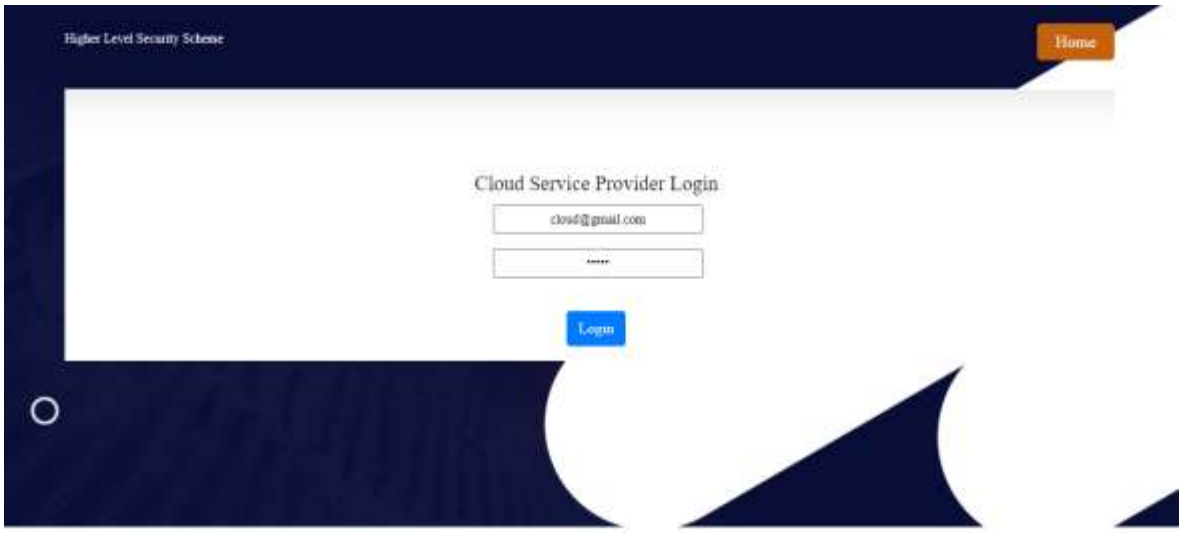

Fig: CSP Login page

Fig: Add Groups

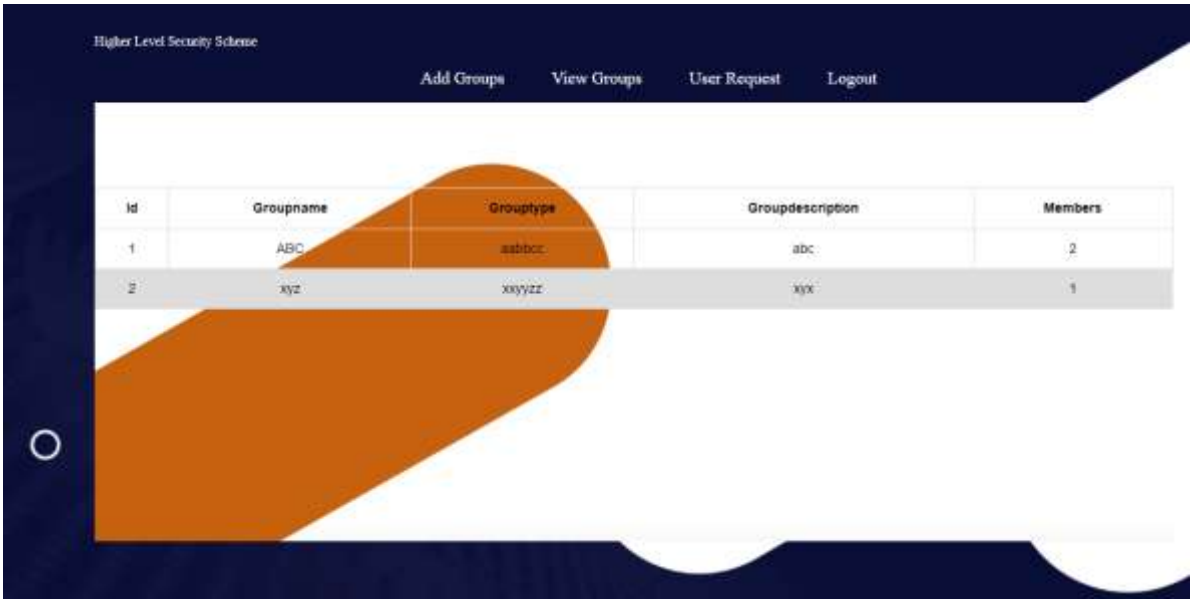

Fig: View Groups

Fig: **User Login**

TEST CASES:

Input	Output	Result
Input text	Tested for whether floods will occur or not	Success

Test cases Model building:

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Read the dataset.	Dataset path.	Dataset need to read successfully.	Dataset fetched successfully.	P
2	Performing pre-processing on the dataset	Pre-processing part takes place	Pre-processing should be performed on dataset	Pre-processing successfully completed.	P
3	Model Building	Model Building for the clean data	Need to create model using required algorithms	Model Created Successfully.	P
4	Floods Prediction	Input Features provided.	Output should be whether floods will occur or not	Model classified floods successfully	P

V. CONCLUSION:

The proposed key access control scheme provides a computationally efficient method for key derivation. The proposed scheme provides both the private cloud security and the functionality, accessibility, and cost savings of the public cloud. With the use of the public cloud by companies, other advantages such as the reliability of the public cloud and the minimum maintenance and management requirements are obtained.

REFERENCES

[1]. S. G. Akl and P. D. Taylor, ''Cryptographic solution to a problem of access control in a hierarchy,'' ACM Trans. Comput. Syst., vol. 1, no. 3, pp. 239–248, 1983.
[2]. C.-C. Chang, R.-J. Hwang, and T.-C. Wu, ''Cryptographic key assignment scheme for access control in a hierarchy,'' Inf. Syst., vol. 17, no. 3, pp. 243–247, May 1992.
[3]. L. Harn and H.-Y. Lin, ''A cryptographic key generation scheme for multilevel data security,'' Comput. Secur., vol. 9, no. 6, pp. 539–546, Oct. 1990.
[4]. M. S. Hwang, C. C. Chang, and W. P. Yang, ''Modified Chang-Hwang-Wu access control scheme,'' Electron. Lett., vol. 29, no. 24, pp. 2095–2096, 1993.
[5]. H. T. Liaw, S. J. Wang, and C. L. Lei, ''A dynamic cryptographic key assignment scheme in a tree structure,'' Comput. Math. Appl., vol. 25, no. 6, pp. 109–114, Mar. 1993.
[6]. S. J. Mackinnon, P. D. Taylor, H. Meijer, and S. G. Akl, ''An optimal algorithm for assigning cryptographic keys to control access in a hierarchy,'' IEEE Trans. Comput., vol. C-34, no. 9, pp. 797–802, Sep. 1985.
[7]. R. S. Sandhu, ''Cryptographic implementation of a tree hierarchy for access control,'' Inf. Process. Lett., vol. 27, no. 2, pp. 95–98, 1988.