International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 02 | Feb-2018 www.irjet.net p-ISSN: 2395-0072

SECURE SCHEME FOR CLOUD-BASED MULTIMEDIA CONTENT STORAGE

NENAVATH THIRUPAL NAIK[1], M VENKATESH NAIK[2], T PADMA[3]

[1]M.TECH SCHOLAR, ST.MARK EDUCATIONAL INSTITUTION SOCIETY GROUP INSTITUTIONS
[2,3] ASSISTANT PROFESSOR, ST.MARK EDUCATIONAL INSTITUTION SOCIETY GROUP INSTITUTIONS

Abstract: *We propose a new design for large-scale multimedia content protection systems. Our design leverages cloud infrastructures to provide cost efficiency, rapid deployment, scalability, and elasticity to accommodate varying workloads. The proposed system can be used to protect different multimedia content types, including 2-D videos, 3-D videos, images, audio clips, songs, and music clips. The system can be deployed on private and/or public clouds. Our system has two novel components:*

(i) Method to create signatures of 3-D videos, and

(ii) Distributed matching engine for multimedia objects.

The signature method creates robust and representative signatures of 3-D videos that capture the depth signals in these videos and it is computationally efficient to compute and compare as well as it requires small storage. The distributed matching engine achieves high scalability and it is designed to support different multimedia objects. We implemented the proposed system and deployed it on two clouds: Amazon cloud and our private cloud. Our experiments with more than 11,000 3-D videos and 1 million images show the high accuracy and scalability of the proposed system. In addition, we compared our system to the protection system used by YouTube and our results show that the YouTube protection system fails to detect most copies of 3-D videos, while our system detects more than 98% of them. This comparison shows the need for the proposed 3-D signature method, since the state-of-the-art commercial system was not able to handle 3-D videos.

1. INTRODUCTION

Cloud computing is using computing resources (hardware and program) that are delivered as a service over a network (commonly the web). The identify comes from the original use of a cloud-formed image as an abstraction for the difficult infrastructure it comprises in system diagrams. Cloud computing entrusts far flung offerings with a user's data, program and computation. Cloud computing consists of hardware and software resources made to be had on the net as managed third-social gathering services. These services most commonly provide entry to evolved application functions and excessive-end network of the server.

1.1 How Cloud Computing Works?

The intention of cloud computing is to use usual supercomputing, or excessive-performance computing power, typically used by military and research facilities, to participate in tens of trillions of computations per 2nd, in client-oriented functions equivalent to financial portfolios, to provide personalized know-how, to furnish information storage or to vigor massive, immersive computer games.

The cloud computing uses networks of colossal groups of servers mostly jogging low-cost patron pc technology with specialized connections to unfold data-processing chores across them. This shared IT infrastructure comprises giant pools of methods that are linked collectively. Commonly, virtualization strategies are used to maximize the power of cloud computing.

1.2 Characteristics and Services Units

The salient characteristics of cloud computing based on the definitions supplied by way of the countrywide Institute of standards and Terminology (NIST) are outlined below:

1.3.1 On-Demand Self-Carrier: A patron can unilaterally provision computing capabilities, similar to server time and community storage, as wanted robotically without requiring human interplay with each provider's provider.

1.3.2 Extensive Network Entry: Capabilities are on hand over the network and accessed through general mechanisms that promote use with the aid of heterogeneous thin or thick client systems (e.g., cellular phones, laptops, and PDAs).

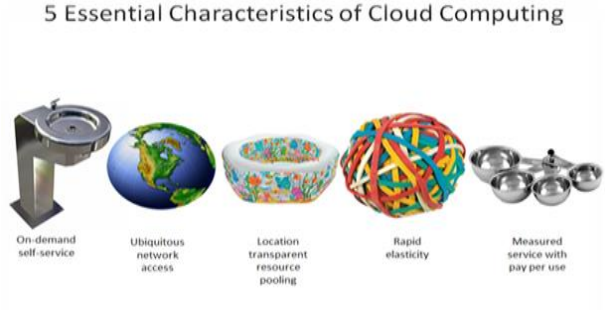

Fig.1 Characteristics of Cloud Computing

2. Proposed Work and Analysis

In this chapter, discussion about Existing System, Disadvantages of the Existing System and Techniques used in Proposed System, Advantages of Proposed System which can have the work that are to be techniques that are possibilities.

2.1 Existing System

Existing techniques have been proposed to protect the data contents privacy via access control. Identity-based encryption (IBE) was first introduced by Shamir, in which the sender of a message can specify an identity such that only a receiver with matching identity can decrypt it.

- Few years later, Fuzzy Identity-Based Encryption is proposed, which is also known as Attribute-Based Encryption (ABE).
- The work by Lewko *et al.* and Muller *et al.* are the most similar ones to ours in that they also tried to decentralize the central authority in the CP-ABE into multiple ones.
- Lewko *et al.* use a LSSS matrix as an access structure, but their scheme only converts the AND, OR gates to the LSSS matrix, which limits their encryption policy to Boolean formula, while we inherit the flexibility of the access tree having threshold gates.
- Muller *et al.* also supports only Disjunctive Normal Form (DNF) in their encryption policy.

2.2 Disadvantages of Existing System

- The identity is authenticated based on his information for the purpose of access control.
- Preferably, any authority or server alone should not know any client's personal information.
- The users in the same system must have their private keys re-issued so as to gain access.
- In this setting, each authority knows only a part of any user's attributes, which are not enough.

2.3 Proposed System

- The data confidentiality, less effort is paid to protect users' identity privacy during those interactive protocols. Users' identities, which are described with their attributes, are generally disclosed to key issuers, and the issuers issue private keys according to their attributes.
- In this propose Annoy Control and Annoy Control-Fallow cloud servers to control users' access privileges without knowing their identity information. In this setting, each authority knows only a part of any user's attributes, which are not enough to figure out the user's identity. The scheme proposed by Chase et al. considered the basic threshold-based KP-ABE. Many attribute based encryption schemes having multiple authorities have been proposed afterwards.
- In our system, there are four types of entities: *N* Attribute Authorities (denoted as *A*), *Cloud* Server, Data Owners and Data Consumers. A user can be a Data Owner and a Data Consumer simultaneously.

2.4 Advantages of Proposed System

- The proposed schemes are able to protect user's privacy against each single authority. Partial information is disclosed in Annoy Control and no information is disclosed in Annoy Control-F.
- The proposed schemes are tolerant against authority compromise, and compromising of up to (N −2) authorities does not bring the whole system down.
- We provide detailed analysis on security and performance to show feasibility of the scheme Annoy Control and Annoy Control-F.
- We firstly implement the real toolkit of a multi authority based encryption scheme Annoy Control and Annoy Control-F

3. 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 and/or finished product it is the process of exercising software with the intent of ensure.

Software system meets its requirements and user expectations and dos not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

3.1 Types of Tests

Test can classify the following types they are:

3.1.1 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.

3.1.2 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.

3.1.3 Functional Testing

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.

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.

3.1.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

3.2 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

3.2.1 Test Strategy and Approach

Field testing will be performed manually and functional tests will be written in detail.

Test Objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to Be Tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

3.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

4. Results

In this chapter the practical interface is discussed. In this chapter the software requirements and the hardware requirements that are necessary to execute the extraction pattern are specified . The graphical user interface that the user or performs while utilizing the essence of bootstrapping algorithm and MLN are discussed.

4.1. System Requirements

Hardware Requirements:

➢ System	:	Pentium IV 2.4 GHz.
➢ Hard Disk	:	40 GB.
➢ Floppy Drive	:	1.44 Mb.
➢ Monitor	:	15 VGA Colour.
➢ Mouse	:	Logitech.
➢ Ram	:	512 Mb.

Software Requirements:

- Operating system : Windows XP/7.
- Coding Language : JAVA
- Tool : Eclipse
- Database : SQL SERVER 2008

4.2 Execution Results:

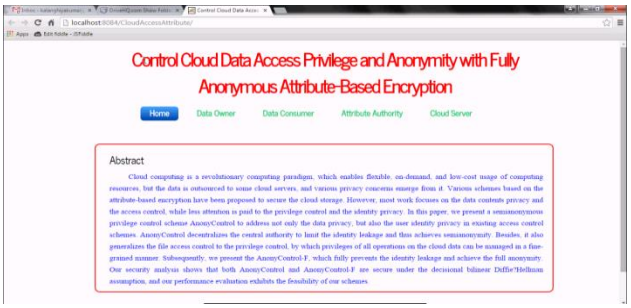

Fig 4 Home Page

The above Fig 4 shows the home page of the cloud data access privilege and anonymity with fully anonymous attribute based encryption.

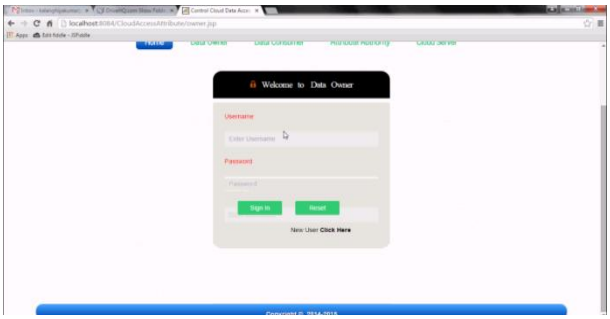

Fig 5 Data Owner Window Login Page

The above Fig 5 shows a form for data owner who can already get a authentication from authority. To enter their user name and password and sign in to cloud for uploading their files.

Fig 6 Data Owner Logged Window

The above fig 6 shows the data owner's logged account, where he can change the password because previously logged by system generated password for the security purpose. After that data owner will request the Attribute authority for uploading the data in the cloud.

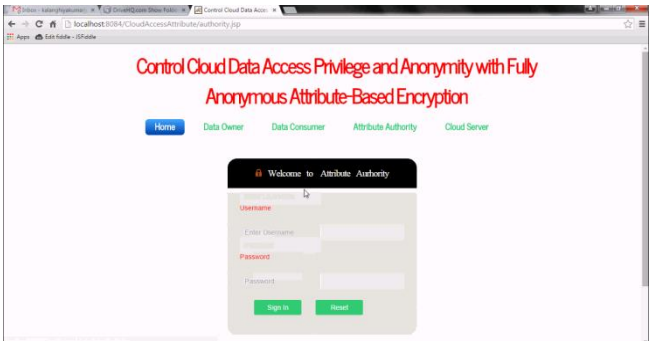

Fig 7 Cloud Access Attribute for Authority

The above Fig 7 shows that the login form for authority who can give the authentication for the Data Owner.

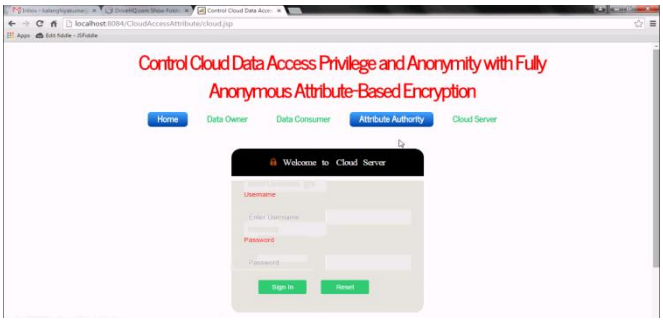

Fig 8 Cloud Server Login Web Page

The above Fig 8 shows that the form for cloud server which can check that the files are in the decrypted format or not.

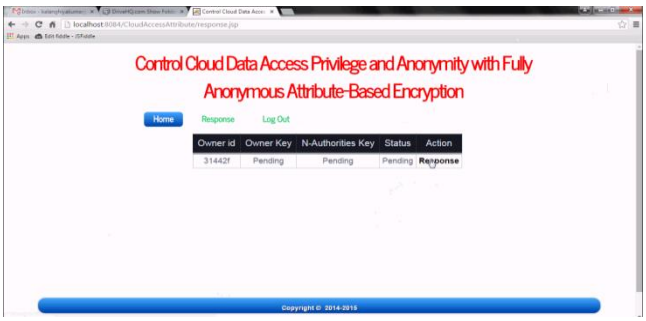

Fig 9 Changing Password for Data Owner

The above Fig 9 shows that if the Data Owner can't have any account in cloud he can register newly in to cloud along with their details like email id ,name role e.t.c .then the cloud can send a public key to their mails like user name and password. If you don't like that password then change it through this form.

Fig.10 Sending Request to Cloud

The above Fig 10 shows that the send a request to the cloud to generate the keys which is through the mails.

Fig 11 Response to Data Owner

The above Fig 11 shows that when the request is pending to data owners then click on response to add the public key in this form. The send the response to the data owner

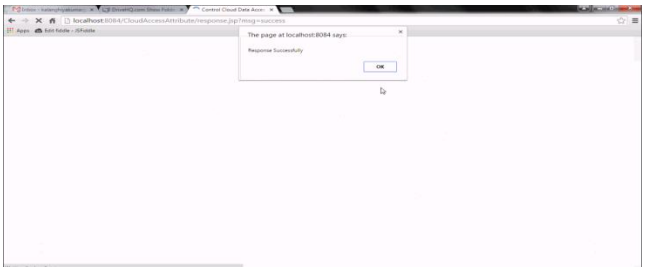

Fig 12 Getting Message

The above Fig 12 shows that after sending response it can get message from cloud that the response will be successful.

Fig 13 Activated Response

The above Fig 13 shows that the cloud can activated response that is status should be granted.

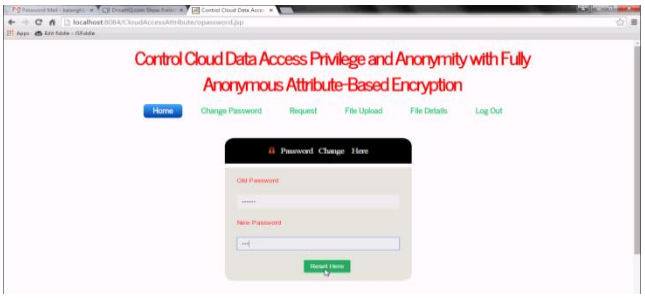

Fig 14 Uploading Files

The above Fig 14 shows that select a file which can uploaded into cloud and give a file id and file name and browse the file where that can be located then click on upload. Then the file must be in encrypted format and it enters in to the cloud using public key.

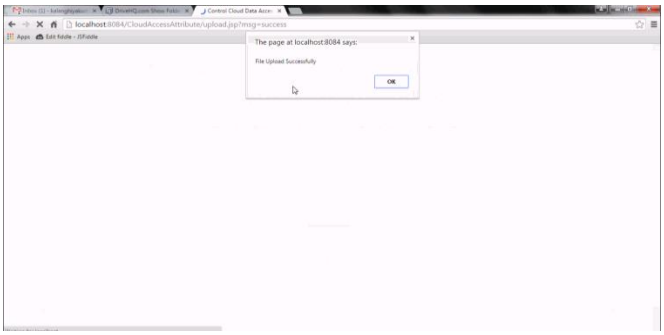

Fig 15 Getting Message

The above Fig 15 shows that after uploading file It can get message from cloud that the upload will be successful.

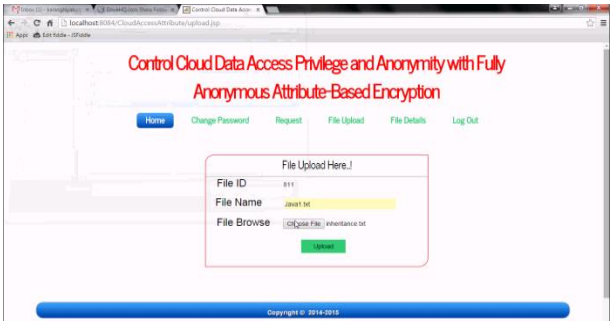

Fig 16 Checking File Details

The above Fig 16 shows that after uploading file it can check whether the file can be uploaded or not using file details. Then it can already enter in the list of uploaded files. These files are in the encrypted format.

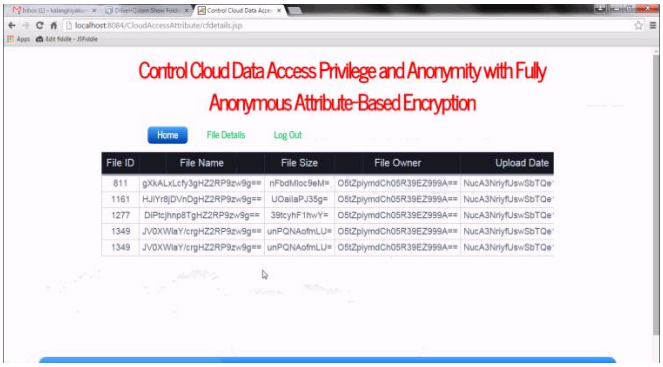

Fig 17 Decrypted Format of a File

The above Fig 17 shows that the user can access the files from data owner by getting two keys from he may want to download such type of files. After downloading that files it can check that the files are in decrypted format.

CONCLUSION

This paper proposes a semi-anonymous attribute-based privilege control scheme AnonyControl and a fully-anonymous attribute-based privilege control scheme AnonyControl-F to address the user privacy problem in a cloud storage server. Using multiple authorities in the cloud computing system, our proposed schemes achieve not only fine-grained privilege control but also identity anonymity while conducting privilege control based on users' identity information.

More importantly, our system can tolerate up to N – 2 authority compromise, which is highly preferable especially in Internet-based cloud computing environment. We also conducted detailed security and performance analysis which shows that Annoy-Control both secure and efficient for cloud storage system. The AnonyControl-F directly inherits the security of the AnonyControl and thus is equivalently secure as it, but extra communication overhead is incurred during the 1-out-of-n oblivious transfer. One of the promising future works is to introduce the efficient user revocation mechanism on top of our anonymous ABE.

Supporting user revocation is an important issue in the real application, and this is a great challenge in the application of ABE schemes. Making our schemes compatible with existing ABE schemes who support efficient user revocation is one of our future works.

References:

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," in Advances in Cryptology. Berlin, Germany: Springer-Verlag, 1985, pp. 47–53.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology. Berlin, Germany: Springer-Verlag, 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 Proc. 13th CCS, 2006, pp. 89–98.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption," in Proc. IEEE SP, May 2007, pp. 321–334.

[5] M. Chase, "Multi-authority attribute based encryption," in Theory of Cryptography. Berlin, Germany: Springer-Verlag, 2007, pp. 515–534.