RESEARCH ARTICLE OPEN ACCESS

Access Control Technique-ABE for Scalable Content Coding in Sharing Networks

V.Maharshi*, A Venkata Pradeep**
**(Master of Technology in Computer Science & Engineering, Vivekananda Institute of Technology & Science, Housing Board Colony, By-Pass Road, Karimnagar , Telangana, India*
Email: maharshi535@gmail.com)
*** (Master of Technology in Computer Science , Vivekananda Institute of Technology & Science, Housing Board Colony, By-Pass Road, Karimnagar , Telangana, India*
Email: a.pradeep5a7@gmail.com)

ABSTRACT
Cloud computing emerged as aspired technology in the last few years. With rapid development in Cloud Computing more and more sensitive data is available for sharing in cloud. On other hand providing security features like access control, encryption and decryption became compulsory to avoid vulnerabilities faced during content sharing. This paper presents a novel access control technique-ABE(Attribute- Based Encryption),which is basically a Cipher text Policy. In Multi- message Cipher text Policy(MCP-ABE), access policy is sent along with cipher text. Now, we propose a technique in which the access policy need not be sent along with ciphertext, we are providing privacy for the encryptor party. Here the encrypting party determines the policy of decryption, while the secret key is associated with a set of attributes in ABE scheme where encryptor specified access policies are hidden. In CP-ABE the cipher text is associated with the access policy. Even the legitimate decrypt or cannot obtain the information about the access policy associated with the encrypted data more than the fact that it can decrypt the data. The scheme is efficient and flexible because MCP-ABE allows a content provider to specify an access policy and encrypt multiple messages with in one cipher text such that only the users whose attributes satisfy the access policy can decrypt the cipher text.

Keywords **-** ABE-Attribute Based Encryption, Access Control, Cloud Computing, Data privacy, Scalable content code.

Date of Submission: 10-11-2017 Date of acceptance: 25-11-2017

I. INTRODUCTION

One of the popular content sharing environments such as social networking are very dynamic in terms of the number of on line users, storage requirement, network bandwidth, computational capability, applications and platforms, thus it is not easy for a service provider to allocate resources following the traditional client-server model. As cloud computing offers application developers and users an abstract view of services that hides much of the system details and inner workings, it is more and more popular in content-sharing applications. However, the weak security provision of cloud computing services is delaying their adoption. As a result, it is imperative for cloud computing based service providers, private or public, to build security functionalities into their services and manage their services following prudent security practices.

Access control is the fundamental security mechanism to facilitate information sharing in a controllable manner. It exerts control over which user can access which resource based on a permission relationship between user attributes and resource attributes, where attributes can be any information deemed relevant for granting access, such as user job function and resource quality, and permission is specified in terms of requirements on the attributes of resource and user. Any user with attributes that meet the requirements has access to that resource.

However, it is challenging to design a suitable access control mechanism in content sharing services due to any individual is able to freely produce any number and any kind of online media such as text, image, sound, video, and presentation any individual is able to grand any access to his media to anyone, at any time an individual may reveal a large number of attributes (e.g. name, age, address, friendship, classmate, fans, hobby, personal interest, gender, and mobility), and some of them can be very dynamics; and individuals may share contents using various devices and bandwidth, and hence demand different access privileges for the same media.

A promising approach to access control in content sharing services is to empower users to enforce access controls on their data directly, rather than through a central administrator. However, this requires flexible and scalable cryptographic key management to support complex access control policies. A naive access control solution is to assign one key for each user attribute, distribute the appropriate keys to users who have the corresponding attributes, and encrypt the media with the attribute keys repeatedly, e.g., the cipher text is produced as c = ε (ε (m, sk1), sk2) to protect message m with attribute key pair (sk1, sk2) and cipher ε (·). This naïve solution is flexible, but it is vulnerable to collusion attack. Technically, a user having key sk1 for one attribute and another user having key sk2 for another attribute can collude to decrypt cipher text c to m. In other words, two users having one attribute each are able to conspire to have the same capability as a user having those two attributes in the naive scheme. Another method is to classify users into different roles based on their attributes, assign role keys to users, and then encrypt the content using the role keys.

II. LITERATURE SURVEY

The most important step in software development process is Literature survey. Determining the time factor, economy and company strength is necessary prior developing the tool. Once these things are satisfied, then next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

2.1 Parallel and Distributed Computing

Distributed systems are groups of networked computers, which have the same goal for their work. The terms “concurrent computing”. “parallel computing, and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly-coupled form of distributed computing, and distributed computing may be seen as a loosely-coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

In parallel computing, all processors have access to a shared memory. Shared memory can be used to exchange information between processors.

In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

The system is represented as a network topology in which each node is a computer and each line connecting the nodes is a communication link. the same distributed system in more detail:
Each computer has its own local memory, and information can be exchanged only by passing messages from one node to another by using the available communication links. The parallel system in which each processor has a direct access to a shared memory.

The situation is further complicated by the traditional uses of the terms parallel and distributed algorithm that do not quite match the above definitions of parallel and distributed systems; see the section Theoretical foundations below for more detailed discussion. Nevertheless, as a rule of thumb, high-performance parallel computation in a shared-memory multiprocessor uses parallel algorithms while the coordination of a large-scale distributed system uses distributed algorithms.

2.2 History

The use of concurrent processes that communicate by message-passing has its roots in operating system architectures studied in the 1960s. The first widespread distributed systems were local-area networks such as Ethernet that was invented in the 1970s.

ARPANET, the predecessor of the Internet, was introduced in the late 1960s, and ARPANET e-mail was invented in the early 1970s. E-mail became the most successful application of ARPANET, and it is probably the earliest example of a large-scale distributed application. In addition to ARPANET, and its successor, the Internet, other early worldwide computer networks included Usenet and Fido Net from 1980s, both of which were used to support distributed discussion systems.

The study of distributed computing became its own branch of computer science in the late 1970s and early 1980s. The first conference in the field, Symposium on Principles of Distributed Computing (PODC), dates back to 1982, and its European counterpart International Symposium on Distributed Computing (DISC) was first held in 1985.

2.3 Applications

There are two main reasons for using distributed systems and distributed computing. First, the very nature of the application may require the use of a communication network that connects

several computers. For example, data is produced in one physical location and it is needed in another location.

Second, there are many cases in which the use of a single computer would be possible in principle, but the use of a distributed system is beneficial for practical reasons. For example, it may be more cost-efficient to obtain the desired level of performance by using a cluster of several low-end computers, in comparison with a single high-end computer. A distributed system can be more reliable than a non-distributed system, as there is no single point of failure. Moreover, a distributed system may be easier to expand and manage than a monolithic uniprocessor system.

Examples of distributed systems and applications of distributed computing include the following:

- **Telecommunication networks:**
- Telephone networks and cellular networks.
- Computer networks such as the Internet.
- Wireless sensor networks.
- Routing algorithms.
- **Network applications:**
- World wide web and peer-to-peer networks.
- Massively multiplayer online games and virtual reality communities.
- Distributed databases and distributed database management systems.
- Network file systems.
- Distributed information processing systems such as banking systems and airline reservation systems.
- **Real-time process control:**
- Aircraft control systems.
- Industrial control systems.
- **Parallel computation:**
- Scientific computing, including cluster computing and grid computing and various volunteer computing projects; see the list of distributed computing projects.
- Distributed rendering in computer graphics.

III. SYSTEM ARCHITECTURE AND DESIGN

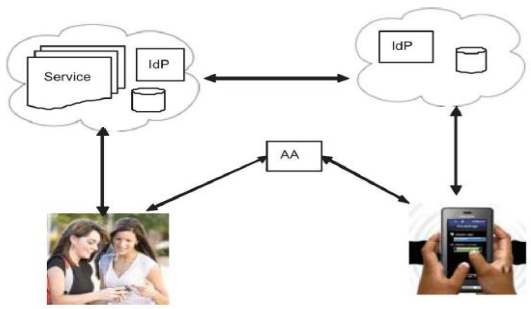

Fig 3.1 System Architecture for Scalable Content Coding in Sharing Networks.

3.1 Existing System

In Existing System, there are multiple owners who may encrypt according to their own ways, possibly using different sets of cryptographic keys. Letting each user obtain keys from every owner who's records she wants to read would limit the accessibility since patients are not always online. An alternative is to employ a central authority (CA) to do the key management on behalf of all record owners, but this requires too much trust on a single authority (i.e., cause the key escrow problem).

Key escrow (also known as a "fair" cryptosystem) is an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys. These third parties may include businesses, who may want access to employees' private communications, or governments, who may wish to be able to view the contents of encrypted communications.

3.1.1 Disadvantages of Existing System:

- In an existing system solution is flexible, but it is vulnerable to collusion attack.
- The existing method is to classify users into different roles based on their attributes, assign role keys to users, and then encrypt the content using the role keys. However, this approach results in high complexity.
- Existing CP-ABE schemes merely deliver one encrypted message per cipher text to all authorized users and are not optimal for efficient sharing of scalable media.

3.2 Proposed System

In proposed System, proposed an information management architecture using CP-ABE and optimized security enforcement efficiency. Furthermore, they employed the architecture and optimization method on two example applications:

An HIPAA (Health Insurance Portability and Accountability Act) compliant distributed file system and a content delivery network.

An approach to access control in content sharing services is to empower users to enforce access controls on their data directly, rather than through a central administrator. However, this requires flexible and scalable cryptographic key management to support complex access control policies. A native access control solution is to assign one key for each user attribute, distribute the appropriate keys to users who have the corresponding attributes, and encrypt the media with the attribute keys repeatedly.

3.2.1 Advantages of Proposed System:

- The present scheme is also secure against user collusion attacks due to use of attribute-based encryption.
- The experiments demonstrate that the present scheme is applicable on smart phone, especially when a cloud platform is available.
- We present an access control scheme for scalable media. The scheme has several benefits which make it especially suitable for content delivery.

IV. MODULES

The media sharing application in cloud environment is composed of the following parties: backend servers, foreground servers, AA, and data consumers (or users).

1) Back end Server

Backend server is part of the infrastructure of the cloud computing platform. According to the National Institute of Standards and Technology (NIST), cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources(e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider's interaction. Cloud computing platforms are assumed to have abundant storage capacity and computation power. Hence, from the viewpoints of network service providers, cloud computing significantly decreases the traffic and storage requirements incurred by their applications.

2) Frontend Server

Foreground server provides the services which are always online. A server is often operated by a cloud service provider (CSP), but sometimes, a user is able to run his/her own services on the cloud platform too. The foreground services may include web service, database service, media maker service, media de-coding service, identity management service, etc.

3) Attribute Authority(AA)

Attribute Authority (AA), a trusted third party, sets up the system parameters of attribute-based encryption system (such as system-wide public key and master key), verifies every user's attributes (e.g., group membership, role, security clearance or authorization information) and issues personal secret key corresponding to the set of attributes of the user. In practice, there could be multiple AAs in a system. For example, a university or corporate may run an AA, and a user may act as an AA for his/her extended family members.

4) User

User may be a data owner, or a data consumer, or both. A data owner produces (protected or unprotected) media content (text, voice, video, etc.), and uploads the media content to cloud servers. To enforce access control to his data, the data owner assigns access privileges to data consumers whom the data owner may or may not know.

A data consumer downloads media con-tent of her interest from cloud servers, and obtains the content based on her attributes and the access policy of the data owner. To this end, the data consumer must obtain from AA a personal secret key SK bound to her set of attributes.

In this data owner-consumer model, the backend servers pro-vide the fundamental platform for storage, networking, etc; the foreground servers provide the interface for media generation, transmission, and computational assistance to users; while AA issues personal secret keys so that access control can be en-forced flexibly based on user attributes and media scalability.

V. SOFTWARE ENVIRONMENT

Features OF. Net

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate.

There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

".NET" is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

The .Net Framework

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).
2. A hierarchical set of class libraries.

The CLR is described as the "execution engine" of .NET. It provides the environment within which programs run. The most important features are

• Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
• Memory management, notably including garbage collection.
• Checking and enforcing security restrictions on the running code.
• Loading and executing programs, with version control and other such features.
• The following features of the .NET framework are also worth description:

Managed Code

The code that targets .NET, and which contains certain extra Information - "metadata" - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

Managed Data

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you're using, impose certain constraints on the features available.

As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn't get garbage collected but instead is looked after by unmanaged code.

Common Type System

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way.

CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn't attempt to access memory that hasn't been allocated to it.

Common Language Specification

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

The Class Library

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary. The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

VI. EXPERIMENTAL RESULTS

Fig 6.1 Garbled text file shown to unauthorized User1 or User 2.

Fig 6.2 Clear text file shown to User 3.

Fig 6.3 Base layer shown to User 4 and User 5.

Fig 6.4 Full video shown to User 6

VII. CONCLUSION

In order to share media content in a controllable manner, a suitable access control mechanism should be deployed. CP-ABE based access control allows a data owner to enforce access control based on attributes of data consumers without explicitly naming the specific data consumers. However, CP-ABE supports only one privilege level and hence is not suitable for access control to scalable media.

In this paper we extended CP-ABE to a novel MCP-ABE and proposed a scheme to support multi-privilege access control to scalable media. As cloud computing is increasingly being adopted and mobile devices are becoming pervasive, the present access control scheme allows a mobile user to offload computational intensive MCP-ABE operations to cloud servers while without compromising user's security. The experimental results indicated that the proposed access control scheme is efficient for securely and flexibly managing media content in large, loosely-coupled, distributed systems. With the assistance of the cloud server, the decryption operation is accelerated significantly at the consumer side. However, the decryption may be still slow for low-end devices because a modular exponentiation operation is required. Thus, one future work is how to speed-up the decryption operation at low-end devices.

REFERENCES

[1] E. Messmer, "Are security issues delaying adoption of cloud computing?," Network World, 27 Apr. 2009 [Online].Available:http://www.networkworld.com/news/2009/042709-burning-secu-rity-cloud-computing.html.

[2] E. Messmer, "Security of virtualization, cloud computing divides IT and security pros," Networkworld.com, 22 Feb. 2010 [Online].

[3] Available:http://www.networkworld.com/news/2010/022210-virtual-ization-cloud security-debate.html.

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

[5] National Inst. Standards and Technol., Secure Hash Standard (SHS), FIPS Publication 180-1, 1995.

[6] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, Sept. 2011.

[7] M.D.Soete, "Attribute certificate," in Encyclopedia of Cryptography and Security,H. C. A. Van Tilborg and S. Jajodia(Eds), ISBN 978-1-4419-5907-2, pp.51, Springer. 2nd ed., Sept. 2011.

[8] G.Ateniese,K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-encryption Schemes with Applications to Secure Distributed 15 Storage," Network and Distributed System Security Symposium, 2005.

[9] S.Yu, C.Wang, K.Ren, andW.Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," IEEE International Conf. on Computer Communications, pp. 1-9, 2010.

[10] Yongdong Wu, Zhuo Wei, and Robert H. Deng "Attribute-Based Access to Scalable Media in Cloud-Assisted Content Sharing Networks".

[11] IEEE Transactions On Multimedia, Vol. 15, No. 4, June 2013.

[12] Changsha Ma and Chang Wen Chen "Attribute-based multi-dimensional scalable access control for social Media sharing".

[13] arXiv:1511.03351v1 [cs.MM] 11 Nov 2015.

V.Maharshi, received his Bachelor of Technology degree in Computer Science and Engineering from Vivekananda Institute of Technology & Science (N6), JNTUH, Telangana, India in 2010,

bagged his Master of Technology in Computer Science and Engineering from Vivekananda Institute of Technology and Science (N6), JNTUH, Telangna, India in November 2014. His research interests include Image processing, Cloud Computing and Network Security.

A.Venkata Pradeep, received Bachelor of Technology degree in Computer Science and Engineering from Vivekananda Institute of Technology and Science, Jawaharlal Technological University, Hyderabad, Telangana, India in 2010 and Masters degree in Computer Science from Vivekananda Institute of Technology & Science, JNTUH, India in December 2013. His research interests include Networking, Ant Net, Traffic Engineering Tasks, Network Security and Cloud Computing.

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

V.Maharshi Access Control Technique-ABE for Scalable Content Coding in Sharing Networks." International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 11, 2017, pp. 01-07.