IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Security and Sharing of Data through Various Access Levels in Transparent Computing

D. Shireesha [1], Ayush Goyal[2], Arun Kumar Jangid [3], Jitesh Jain [4]

Associate Professor, Computer Science and Engg, Guru Nanak Institutions Technical Campus, Hyderabad, India[1]

B.Tech Student, Computer Science and Engg, Guru Nanak Institutions Technical Campus, Hyderabad, India[2,3,4]

Abstract: Transparent computing, the hosts and the users are working on a low-cost high-performance environment but they are placed at distinct places at distinct locations due to which their accessibility decreases. In this paper, we have introduced various levels of accessing the data and sharing them. In these levels, each level will provide a security check to the user and gives restricted access. This proposed system which we were shown is effective in providing security at various levels, providing accessibility to data from anywhere and protecting it from different hazardous attacks.

Keywords: Transparent computing, Authentication, data access, security.

I. INTRODUCTION

With increase in the network technologies computing paradigms have helped many of the users. Pervasive computing helps to provide services anytime, anywhere and by any means. But it has disadvantages with its inability to perform on different kinds of operating system. So, to overcome such difficulties transparent computing came into existence.

This was developed by extending von-Neumann architecture spatio-temporally. Transparent computing[1], [4], [5] brings the concept of cloud computing which helps the users to concentrate on data without able to carry out the physical device whenever the information is needed. In this system, a separate OS is made for cloud called as Trans OS is used in which all the codes of the operating system are stored on a central server through which it can be accessed on any kind of server let it be windows or Linux.

Information security has been greatly improved [2], [3]. The user’s data is protected from theft and change in the data through centralized management at the servers where the data has been stored by the users. If a scenario is visualized of deploying transparent computing in an organization, the information revealed to the members in the organization may have different access permissions like some users may only read the file but could not edit it, some may both read and edit the file etc.

In transparent computing the data and the results are to be stored in the transparent servers (TS) so there may be a possibility that an unauthorized user may access the data stored in the server without the prior knowledge of the user. Hence it is necessary to encrypt system which allows conversion of data of the users to cipher text which is to be understandable by the users who are try to share the after decrypting it with the help of a key.

Some encryption schemes consume lot of resources for their encryption such as Attribute Based Encryption (ABE) which enables it into less efficient. Moreover, to provide efficient data sharing has become a problem. Hence, in this paper, the sharing of resources and accessing of data is provided through certain modules which includes an Authentication Server (AS). It adds the data owner, providing the data owner only to access the given files.

II. RELATED WORKS

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

A. Spatio-temporal Extension

With the rapid improvements in hardware, software and networks, the computing paradigm has also shifted from mainframe computing to ubiquitous or pervasive computing, in which users can get their desired services anytime, anywhere and any means. However, the emergence of ubiquitous or pervasive computing has brought many challenges, one of which is that it is hard to allow users to freely obtain desired services, such as heterogeneous OSes and applications via different light-weight embedded devices. We have proposed a new paradigm by extending the von Neumann architecture spatio-temporally, called Transparent Computing, to store and manage the commodity programs including OS codes centrally, while streaming them to be run in non- state clients. This leads to a service-centric computing environment, in which users can select the desired services on demand, without concerning these services' administrations, such as their installation, maintenance, management, upgrade, and so on.

DOI10.17148/IJARCCE.2017.63189

B. TransOS

Cloud computing has become a hot topic recently. Among these research issues, cloud operating systems have attracted extensive attention. However, to date, there is no answer to such issues as what a cloud operating system is and how to develop one. This paper proposes a cloud operating system, TransOS, from the viewpoint of transparent computing, in which all traditional operating system codes and applications are centrally stored on network servers, and an almost bare terminal dynamically schedules the necessary codes selected by users from the network server, and runs them mostly with the terminal's local resources. The TransOS manages all the resources to provide integrated services for users, including traditional operating systems. This paper first introduces the concept of transparent computing as a background and presents TransOS and its main characteristics. It then gives a layered structure-based designation of TransOS and finally illustrates one example of its implementation.

C. Information Security

The rapid development of computer network technologies and social informationalization has brought many new opportunities and challenges in information security. With improved information and service sharing enjoyed by more and more people, how to strengthen the information security has become an increasingly critical issue. In this paper, we propose a new network security mechanism based on a novel computing paradigm, i.e., transparent computing, which is based on the extended von Neumann architecture.

This paradigm separates the program storage and execution, which is implemented in the network environment. It is realized by a new generation server and client BIOS, namely EFI BIOS, and coordinated with the MetaOS management platform and related switching and input/output devices of transparent computing.

D. Hierarchical Attribution Encryption

With rapid development of cloud computing, more and more enterprises will outsource their sensitive data for sharing in a cloud. To keep the shared data confidential against untrusted cloud service providers (CSPs), a natural way is to store only the encrypted data in a cloud. The key problems of this approach include establishing access control for the encrypted data, and revoking the access rights from users when they are no longer authorized to access the encrypted data.

This paper aims to solve both problems. First, we propose a hierarchical attribute-based encryption scheme (HABE) by combining a hierarchical identity-based encryption (HIBE) system and a ciphertext-policy attribute-based encryption (CP-ABE) system, so as to provide not only fine-grained access control, but also full delegation and high performance. Then, we propose a scalable revocation scheme by applying proxy re-encryption (PRE) and lazy re-encryption (LRE) to the HABE scheme, to efficiently revoke access rights from users.

III. SYSTEM MODEL

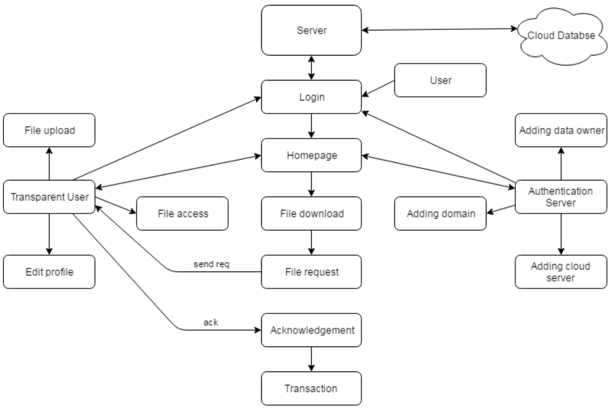

Fig. 1. System Model

In our system, we have a transparent user which sends the appropriate acknowledgement after it receives the request from authenticated user. Transparent user can also perform operations like editing the profile, uploading of file. Authentication Server is used to verify the valid user and check and its accessibility permissions and can add a domain, sub domain. After the user receives the acknowledgement from the transparent user the user is now able to download the file and perform appropriate operations.

IV. PROPOSED SCHEME

A. Modules

1) User Interface Design:

This is the first module for our paper. In this User Interface Design, we create Registration and Login Page, If you are a new user go to registration page and register your own account, After Registration you will go to login page and login your account, After the login page you will get your own account.

2) Authentication Server:

This is the main module of our paper which acts as admin for our entire project. It has several operations likes including of a cloud server, data owner, domain and sub domain.

3) Transparent Server:

This is the third module of our paper after verifying the user, login of the TS page will be taken directly to its home page, which consists of details of the user, uploading of the file in real time cloud, details of file, control access and transactions.

4) User:

In this final module after login user will be taken to the home page, for more safety a distinctive key is provided and only after entering the key the user is able to enter the login page. The user can download the file, send a request to transparent server for access permission and after receiving acknowledgement perform transactions.

DOI10.17148/IJARCCE.2017.63189

V. ALGORITHM

The algorithm of Diffie and Hellman is that it is simple to calculate powers modulo a prime but it is difficult to do reverse of that process: If anyone asks which power of 2 modulo 17 is 9, we must calculate a lot to get the answer, even though 17 is a small prime. If we use a large prime instead, then this becomes a very difficult problem even on a computer.

Steps
1. Subramaniyam and krishnamacharyulu, using unsafe connection, agree on a huge prime r and s generator d. They don't care if someone tries to listen it.
2. Subramaniyam chooses some large random integer us < r and keeps it secret. Likewise krishnamacharyulu chooses uk < r and keeps it secret. These are their "private keys".
3. Subramaniyam computes his "public key" vs dus (mod r) and sends it to krishnamacharyulu using unsafe connection. Krishnamacharyulu computes his public key vk duk and sends it to subramaniyam. Here 0 < vs < r, 0 < vk < r.
As already mentioned, sending these public keys with unsafe connection is safe because it would be too hard for someone to compute us from vs or uk from vk, just like the powers of 2 above.
4. Subramaniyam computes ws vkus (mod r) and krishnamacharyulu computes wk vsu k (mod r). Here ws < r, wk < r.
But ws = wk, since ws vku s (duk)us = d(us ak) (mod r) and similarly wk (du s)uk = d(us uk) (mod r). So this value is their shared secret key. They can use it to encrypt and decrypt the rest of their communication by some faster method.
In this calculation, notice that the step vkus (duk)us involved replacing d uk by its remainder vk, (in the reverse direction) so we were really using the "as often as you want" principle.

VI. PERFORMANCE ANALYSIS

A. User Interface Design
Input : Login name and Password.
Outpu t: If Valid admin Open the admin window otherwise error page.

B. Authentication Server
Input: Authentication Server is nothing but like admin
Output : Authentication Server add all details and checking.

C. Transparent Server
Input : Transparent Server is login and user details.
Output : User details, file upload, file access controls maintain

D. User
Input : User download, send request
Output : User information with help of security key it maintain, File Access Control. The version of this template is V2. Most of the formatting instructions in this document have been compiled by Causal

VII. CONCLUSION

In this paper, a different variety of security is provided with many security levels. Our goal is to provide security to the user data access, cloud, communication, and transactions in transparent computing. The proposed scheme is very effective in enabling efficient data sharing, data storage and data accessibility.
In our future work, we are going to improve our theme by deploying multiple ASs to avoid the potential bottleneck between the users and also the TSS, and make sure the high availableness of the system.

REFERENCES

[1]. Tao Peng, Qin Liu, Guojun Wang ,"A Multilevel Access Control Scheme for Data Security in Transparent Computing "
[2]. Y. Zhang and Y. Zhou, “TransOS: a Transparent Computing-based Operating System for the Cloud,” International Journal of Cloud Computing, vol. 1, no. 4, 2012, pp. 287–301.
[3]. Y. Zhang, L. T. Yang, Y. Zhou, and W. Kuang, “Information Security Underlying Transparent Computing: Impacts, Visions and Challenges,” Web Intelligence and Agent Systems, vol. 8, no. 2, 2010, pp. 203–217
[4] Y. Zhang and Y. Zhou, “Transparent Computing: Spatio-temporal Extension on Von Neumann Architecture for Cloud Services,” Tsinghua Science and Technology, vol. 18, no. 1, 2013, pp. 10–21
[5]. Y. Zhang and Y. Zhou, “Transparent Computing: a New Paradigm for Pervasive Computing,” Ubiquitous Intelligence and Computing:2006 International Conf. (UIC 06), 2006, pp. 1–11.
[6]. G. Wang, Q. Liu, J. Wu, and M. Guo, “Hierarchical Attribute-based Encryption and Scalable User Revocation for Sharing Data in Cloud Servers,” Computers & Security, vol. 30, no. 5, 2011, pp. 320–331.
[7]. Y. Zhang and Y. Zhou, “4VP: a Novel Meta OS Approach for Streaming Programs in Ubiquitous Computing,” Advanced Information Networking and Applications: 21st International Conf. (AINA 07), 2007, pp. 394– 403.
[8]. G. Wang, Q. Liu, Y. Xiang, and J. Chen, “Security from the Transparent Computing Aspect,” Pro. 2014 IEEE Conf. Computing, Networking and Communications (ICNC), 2014, pp. 216–220
[9]. Q. Liu, G. Wang, and J. Wu, “Time-based Proxy Re-encryption Scheme for Secure Data Sharing in a Cloud Environment,” Information Sciences, vol. 258, 2014, pp. 355–370