Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 4, Issue 5, May-2017

Data Sharing Management in Transparent Computing

Shashank Bapure ,Shubham Mate, Shubham Poman , Pankaj Patil ,Archana Gaikwad

Computer Engineering, Dr D.Y.Patil School Of Engineering , Lohegaon ,Pune,India.

Abstract-

The transparent computing is gaining more and more courtesy in recent times. Resources are stored on remote servers, and delivered on demand to clients in a streaming way. The centralized management at servers can bring convenience security for user's information. This approach is greatly useful to protect the user data with different security levels, and provide multilevel access control and valid identity authentication. The proposed scheme is effective in multilevel data security, flexible in authorized resource sharing. Scope of Transparent Computing is enormous and wide spread, which can be used in almost all the information technology domains. By asymmetric group key agreement and group signature, propose an efficient data auditing scheme for to check the file Integrity while at the same time providing some new features, such as traceability and countability.

Keywords *- Access control , Transparent computing, Authentication, Computer security Multilevel security, Access control, Privacy.*

I. INTRODUCTION

Transparent computing is one of the emerging technologies, which allows users to enjoy user-controlled services by extending the stored program concept in the von Neumann architecture into the networking environments spatio-temporally. Transparent computing loads a variety of heterogeneous OSs and applications dynamically on different device. This feature enables users to focus on the available application services without caring about which physical device will be used and what OS should be run on it. The new concept comes with many advantages in information security aspect. The centralized management at the server side have a security related many advantages. By using this type of security the user data makes a secure, and also reduce the risk of data theft and the risks of information leakage. But it is challenging in provide a security in the different Oss, data and applications are centralized in serves and they are shared by different users in transparent computing system.

In this paper we provide three types of users. Public users which are can access the files which are public files. These are encrypted by the owner and give access to the all users. A second type of users, in this type of users, the user can access the file that has a encrypted by the owner and give access to the specific users. And third type of user in this the file is encrypted by owner and could not give access to the any user. Only the owner can access the file and performs read, write what he/she wants to perform. In this AS is the main part. This is going to authenticate the user. And also checks the permission of the user related to the file. The AS can validate the user by using a username and password.

II. LITERATURE SURVEY

i. Transparent Computing: Analysis and a Case Study of Transparent Computing Implementation with UEFI:

Increase the tread research and technological more advances; the pervasive computing is emerging rapidly as an exciting new discipline to provide computing and communication services all the time and everywhere in data sharing. After a comparative analysis on traditional paradigms in transparent computing, that is not a user friendly, users can not get services from computer easily and is one of the main reasons. Transparent Computing will be presented to solve this problem partially. But also it is not give the complete solutions. In this papers also present some primitive real and experimental results to show that it is a feasible and efficient solution for future computing infrastructure.

Disadvantages 1) Also it is not give the complete Solutions for transparent computing.

ii. TransOS: a Transparent Computing-based Operating System for the cloud computing:

Computing has become a hot topic recently. Among these research issues, 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.

Disadvantages:

a. Computing is impossible if you cannot connect to the Internet.
b. A dead Internet connection means no work and in areas where Internet connections are few or inherently unreliable, this could be a deal-breaker.

iii. Hierarchical Attribute-based Encryption and Scalable User Revocation for Sharing Data in Cloud Servers:

To keep the shared data confidential against any kind of misuse, a natural way is to store only the encrypted data in a applications server. The key problems of this approach include establishing access control for the encrypted data. Previous Attribute -Based Encryption systems used attributes to describe the encrypted data. While in our system attributes are used to describe a user's credentials, and encrypting data determines a policy for who can decrypt. However, when organization users outsource confidential data for sharing on Application servers, the adopted encryption system should not only support fine-grained access control, but also provide high performance, practicability, and scalability to best serve the

needs of accessing data anytime and anywhere. This paper, proposes a scheme to help the organization to efficiently view and access confidential data on Application servers. We achieve this goal by first combining the hierarchical identity-based encryption (HIBE) system and the CP-ABE system. The proposed scheme not only achieves scalability due to its hierarchical structure, but also inherits flexibility and fine-grained access control in supporting compound attributes of ASBE.

Disadvantages:

a. Stored data might not be secure.

b. Performance is slow.

III. IMPLEMENTATION DETAILS

Implementation Details:

We aim to build a prototype system which is shown in Figure, which is used for authenticate the user and provide a Multi Level security to the files for reading, writing and updating in transparent computing. With Public Auditing on the server regarding client access information that Increase the user Efficiency.

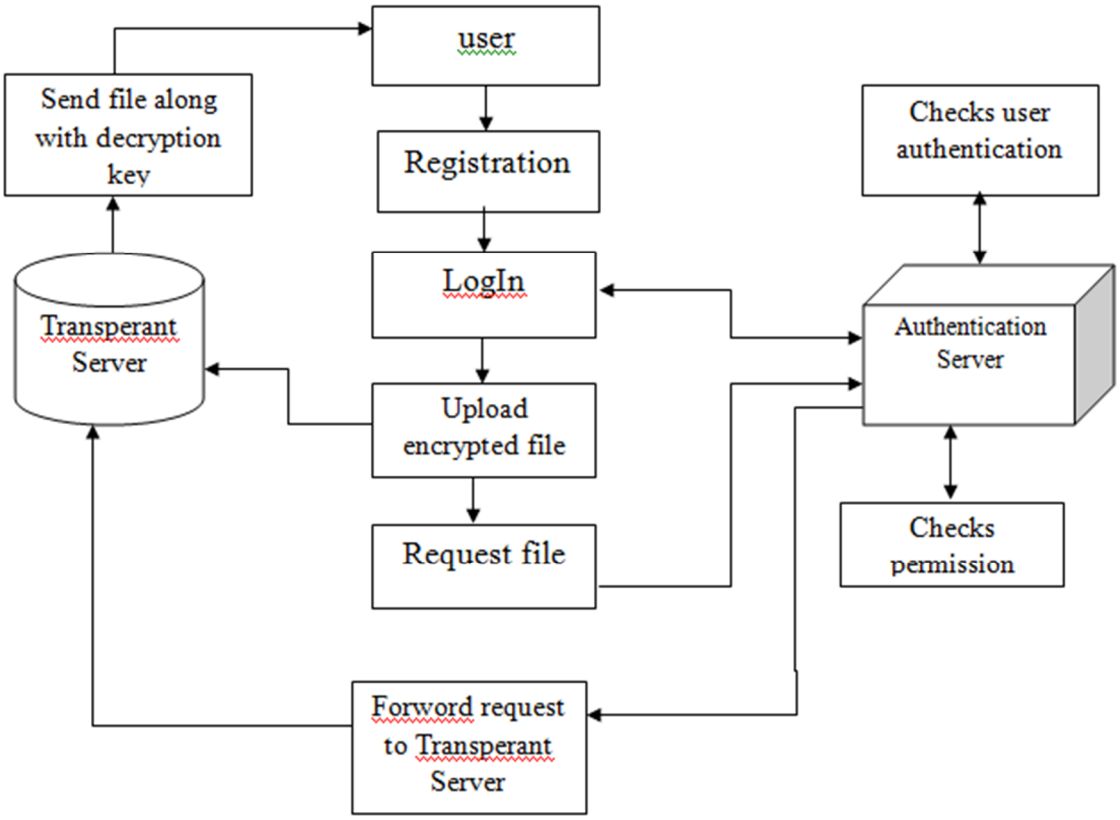

Fig 1. System Diagram

As shown in figure,

User:

The user is registered first. After that the user is login into the system. At the time of login Authentication server performance the authentication of the user. Once the login success then user will upload the encrypted files and also make a request for file. And get the requested file along with decryption key. Also we send the decryption key of file on user mail and opt also used for Authentication Purpose.

Authentication Server:

It is performed the authentication of the user. It is also checks the permission of the user related to the file. After it forwards the request to transparent Server. And the History Of all data user is maintained.

Transparent Server:

It sends the requested file along with decryption key to the user.

IV. Algorithm used

ECC : Elliptical curve cryptography (ECC) is a public key encryption technique based on *elliptic curve theory* that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers. The technology can be used in conjunction with most public key encryption methods.

1.select the file type then select plain text from the file

2.After selecting file select the output file

3.After selecting output file check if file compress or not

4.if the file compress then check the plain text is converted to cypertext or not(encrypted file)

5.if text in file are hidden or converted to cypertext then encryption is successful.

6.for retrieving encrypted, hidden, compressed message select the output file for retrieving output file enter key or password.

Key generation:-

parameters (q, FR, a, b, G, n, h).

1. Select a random number d, d ∈ [1, n – 1]

2. Compare Q = dG.

3. public key is Q and private key is d.

A public key Q = (xq, yq) associated with the domain parameters (q, FR, a, b, G, n, h) is validated using the following procedure

1. Check that $Q \neq O$

2. Check that xq and yq are properly represented elements of Fq

3. Check if Q lies on the elliptic curve defined by a and b.

4. Check that $nQ = O$

2. SHA1 algorithm: Secure Hash Algorithm it is used for the Public Auditing of file to check the integrity.

V. DESIGN SCREENSHOTS:

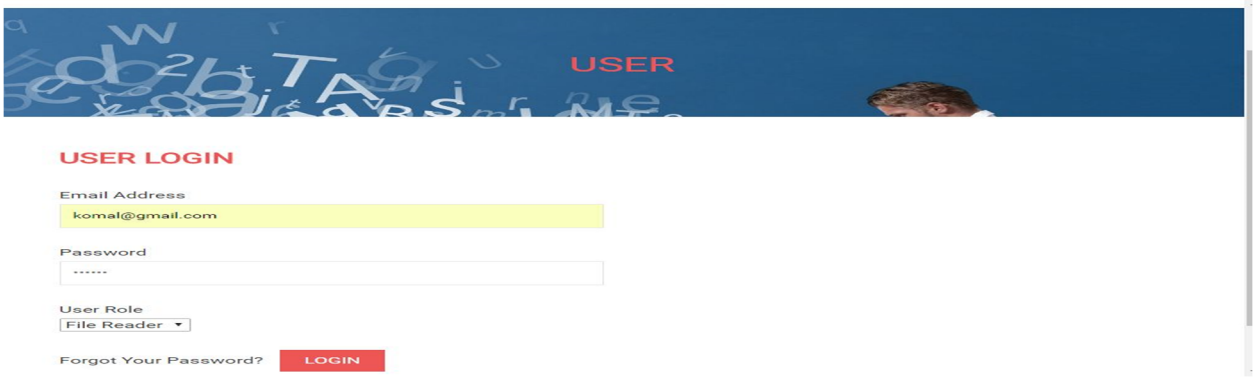

Fig 1.User Login

User Login: This is User Login page. There are three types of Users: Creater, Reader and Writer. First User Registered then Login.

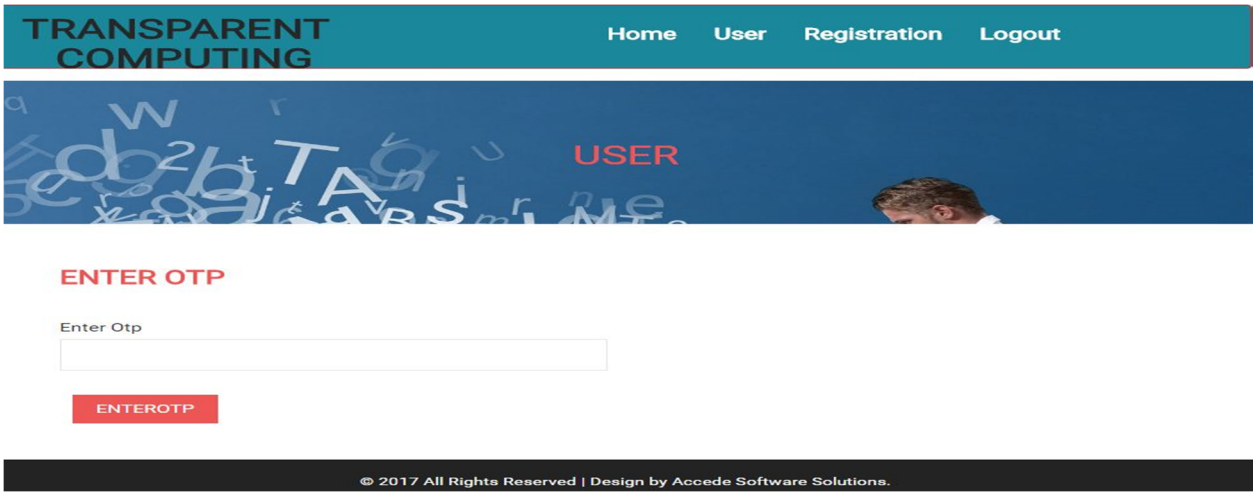

Fig 2. OTP

OTP Password: This is OTP Login page. After User Login enter the correct otp password then login successfully. OTP Concept is used to security pupose.

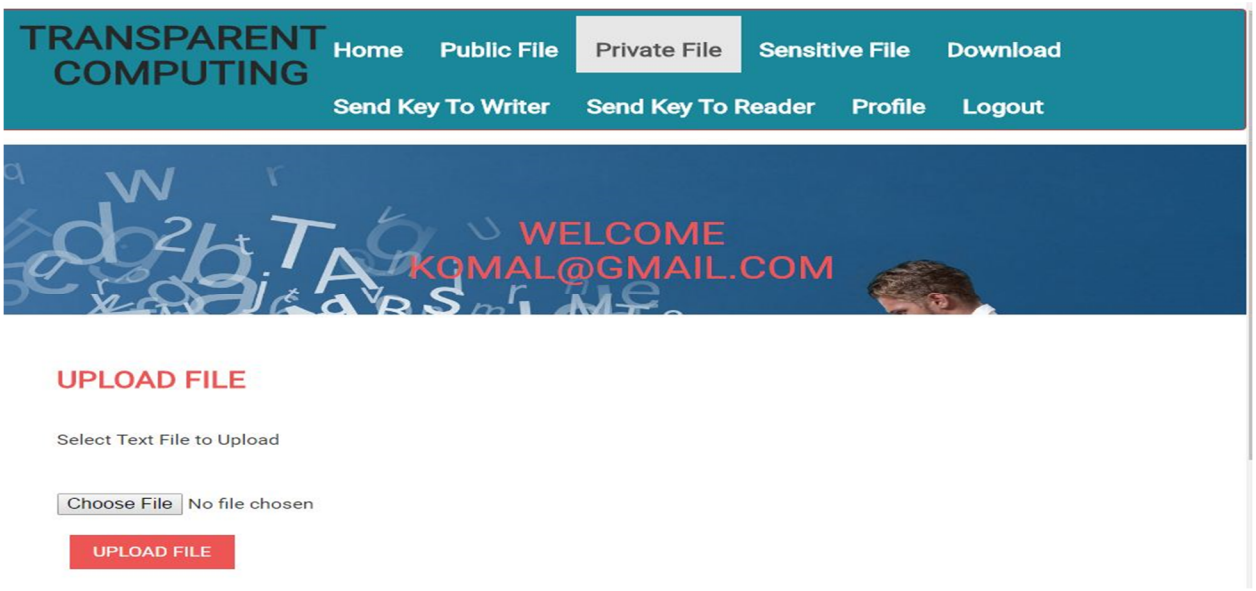

Fig3.Upload Private File

Upload Private File: This is Private File Uploading page. In this page Creater Upload the Private File.

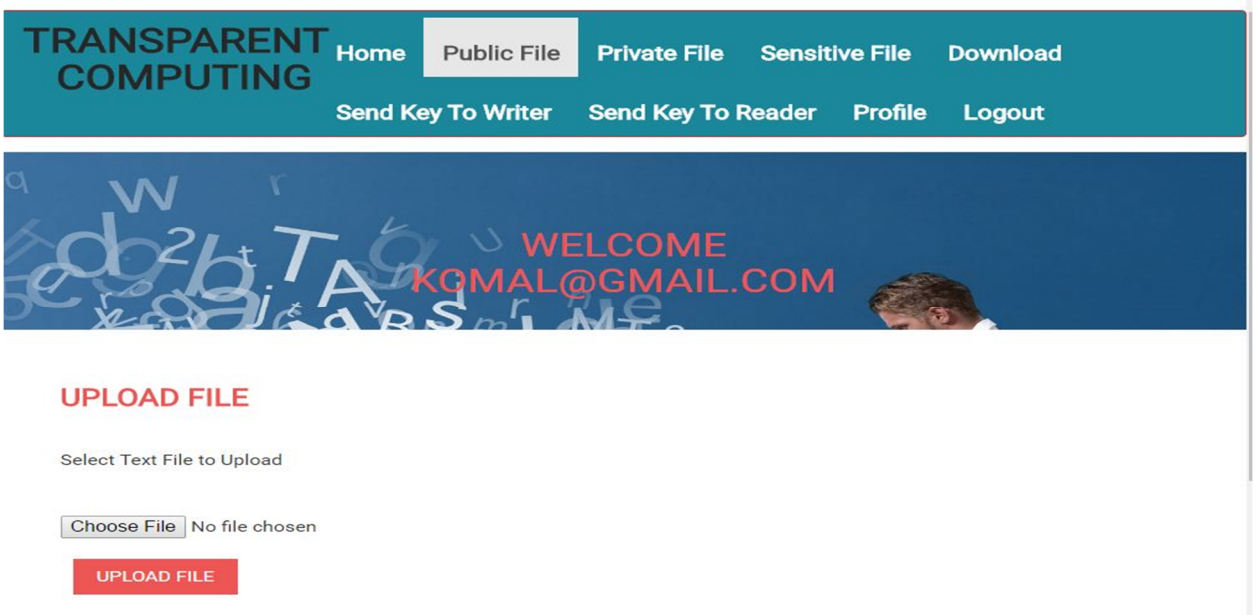

Fig4.Upload Public File

Upload Public File: This is Public File Uploading page. In this page Creater Upload the Public File.

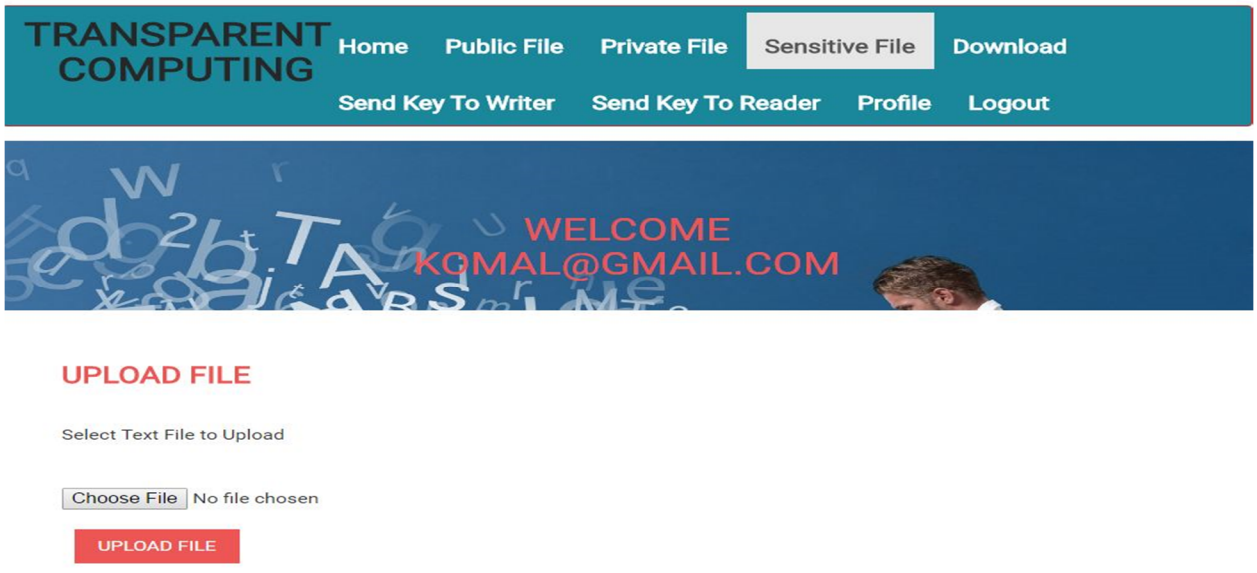

Fig5.Upload Sensitive File

Upload Sensitive File: This is Sensitive File Uploading page. In this page Creater Upload the Sensitive File.

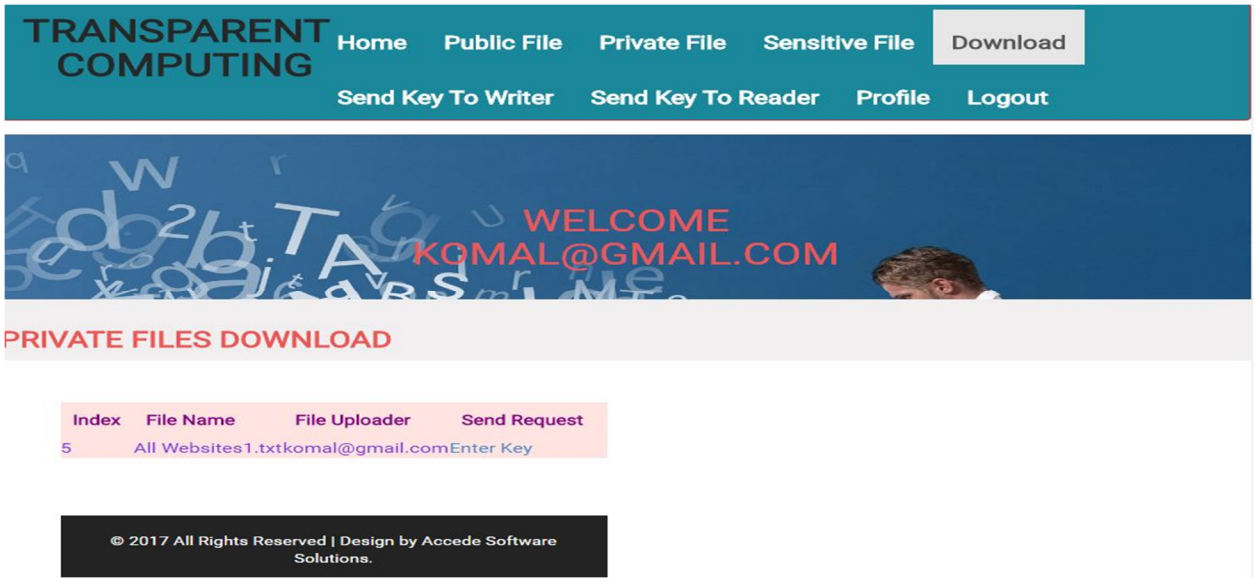

Fig6. Download Private File

Download Private File: This is Download Private File Page. In This Page Creater Downloads the only private files. Other Files not downloaded the User Creater.

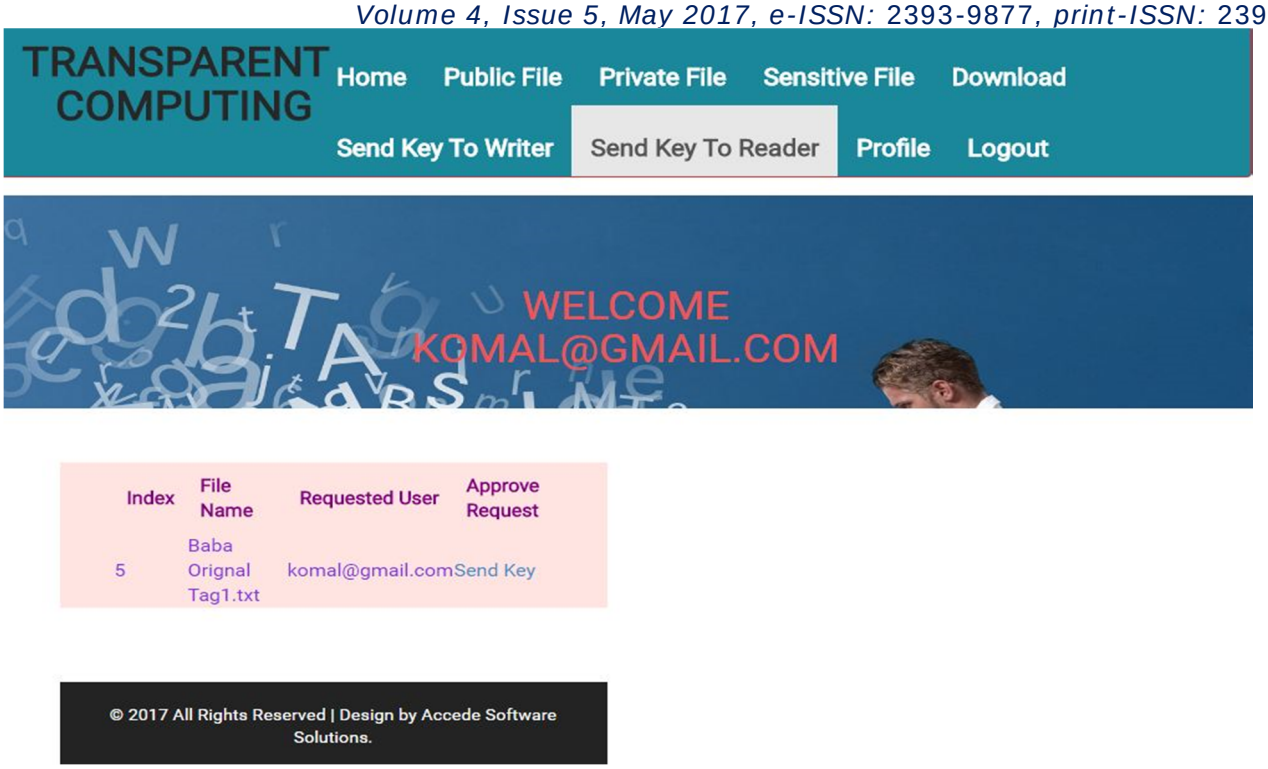

Fig7. Send Key To Reader.

Send Key To Reader: This is Send Key To Reader Page. Reader Download sensitive files. But, without encryption key downloading is not possible then Creater sends the key to Reader for Downloading Sensitive file.

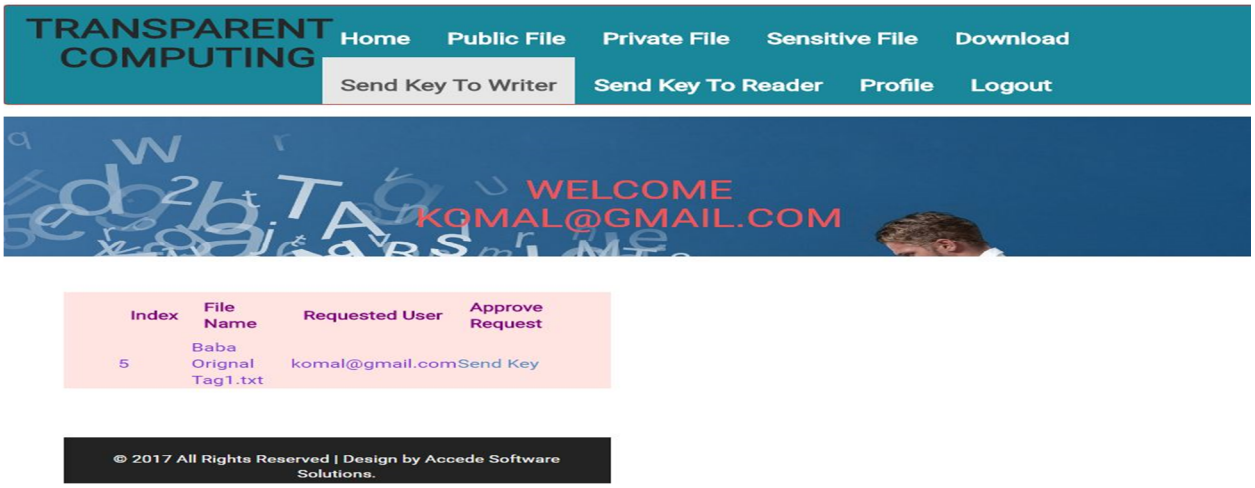

Fig8. Send Key To Writer.

Send Key To Writer: This is Send Key To Writer Page. Writer Download sensitive files. But, without encryption key downloading is not possible then Creater sends the key to Writer for Downloading Sensitive file.

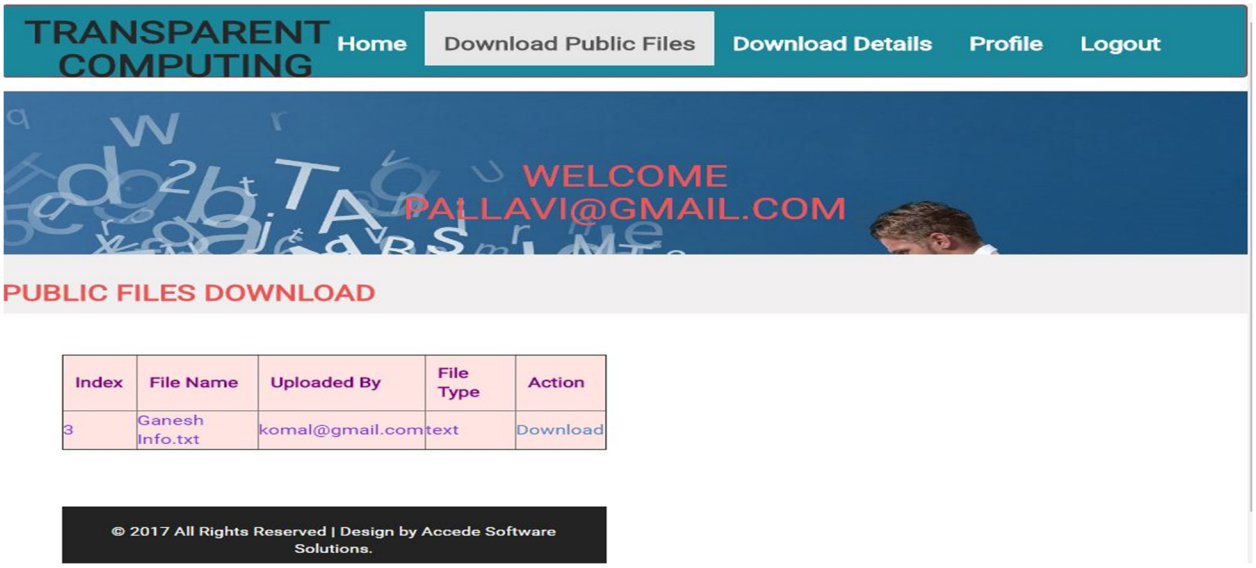

Fig9. Download Public Files.

Download Public Files: This is Download Public Files Page. Reader Downloads only for public files. Public Files are downloaded without encryption key.

Fig10. Download Details of Public Files.

Download Details Of Public Files: This is Download Details of Public Files Page. In this Page show the all details related to public files.

Fig11. Search Sensitive File.

Search Sensitive Files: This is Search Sensitive Files Page. In this Page search the sensitive file and send the download request to creater. After Send Download Request, Creater Sends the Encryption key to Writer.

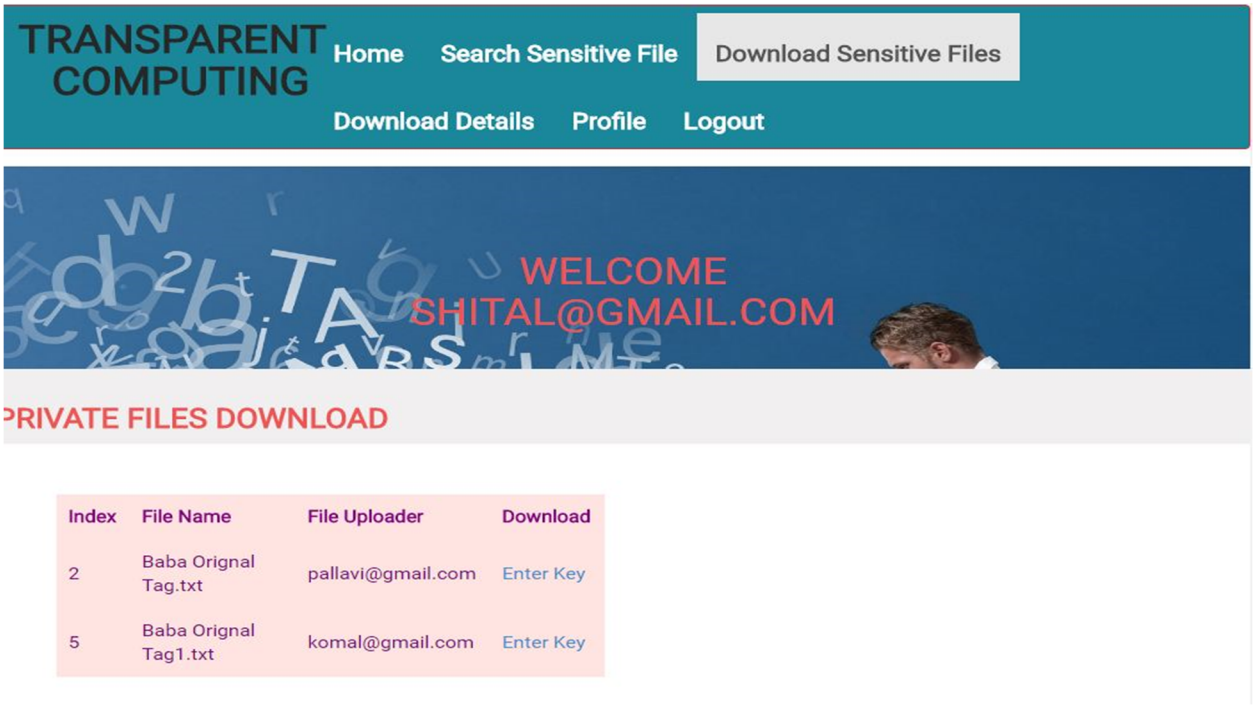

Fig12. Download Sensitive File.

Download Sensitive Files: This is Download Sensitive Files Page. In this Page Enter the Encryption key and downloads the specified file.

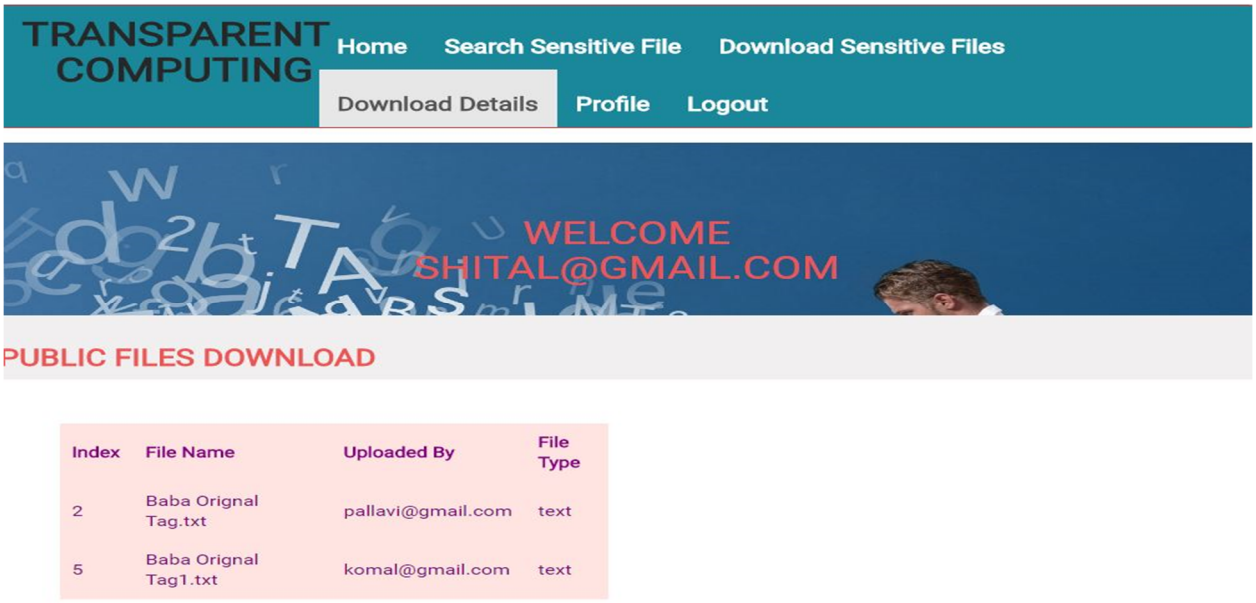

Fig13. Download Sensitive File Details.

Download Sensitive File Details: This is Download Sensitive File Details Page. In this Page Show the all Sensitive file related Details.

VI. CONCLUSION

In Our Proposed system we have implemented multilevel security for the access to the files. The owner of the file gives access permission to the users. The proposed scheme is effective in multilevel data security, flexible in authorized resource sharing Public Auditing, and now days Scope of Transparent Computing is enormous and wide spread, which can be used in almost all the information technology domains.

In Future you can used more Secure Algorithm and authentications Method that Can increase the multilevel security, access control, Public auditing and all.

VII. REFERENCES

1. A Multilevel Access Control Scheme for Data Security in Transparent Computing, 2016 IEEE.

2. Y. Zhang and Y. Zhou, Transparent Computing: Spatio-temporal Extension on Von Neumann Architecture for Services, Tsinghua Science and Technology, vol. 18, no. 1, 2013, pp. 1021.

3. Y. Zhang and Y. Zhou, Transparent Computing: a New Paradigm for Pervasive Computing, Ubiquitous Intelligence and Computing:2006 International Conf. (UIC 06), 2006, pp. 111.

4. 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. 287301.

5. Y. Zhang, L. T. Yang, Y. Zhou, and W. Kuang, Information Security Underlying Transparent Computing: Impacts, Visions and Challenges,Web Intelligenceand Agent Systems, vol. 8, no. 2, 2010, pp. 203217.

6. 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. 216220.

7. 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. 355370.

8. G. Wang, Q. Liu, J. Wu, and M. Guo, Hierarchical Attribute-based Encryption and Scalable User Revocation for Sharing Data in Cloud Servers, Computers and Security, vol. 30, no. 5, 2011, pp 320331.

9. 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.

10. H.-A. Park, J. W. Hong, J. H. Park, J. Zhan, and D. H. Lee, Combined Authentication-based Multilevel Access Control in Mobile Application for Dailylifeservice, IEEE Transactions on Mobile Computing, vol. 9, no. 6, 2010, pp. 824837.