International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 2, Issue 4, April 2014)

An Information Security Scheme for Cloud based Environment using 3DES Encryption Algorithm

Shaunak S.Ganorkar[1], Shilpi U.Vishwakarma[2], Sagar D.Pande[3]
[1,2,3]*Undergraduate, P.R. Patil College of Engineering & Technology, Amravati University*

***Abstract*— Cloud computing is the apt technology for the decade. It allows user to store large amount of data in cloud storage and use as and when required, from any part of the world, via any terminal equipment. While Cloud services offer flexibility, scalability and economies of scale, there have been commensurate concerns about security. On the similar terms, we have chosen to make use of a combination of authentication technique and key exchange algorithm blended with an encryption algorithm. In this project, we have proposed to make use of 3DES algorithm which is a well-known symmetric cryptosystem and is widely used for secure data transmission, along with that we will blend it with Random Key Generator and Graphical Password to add an extra security measure. This proposed architecture of three way mechanism and the use of symmetric method of encryption make it tough for hackers to crack the security system, thereby protecting data stored in cloud. This Cipher Block Chaining system is to be secure for clients and server. The security architecture of the system is designed by using DES cipher block chaining, which eliminates the fraud that occurs today with stolen data. There is no danger of any data sent within the system being intercepted, and replaced. The system with encryption is acceptably secure, but that the level of encryption has to be stepped up, as computing power increases. Results in order to be secure the system the communication between modules is encrypted using symmetric key**

***Keywords*— Cloud, Cloud Security, Cryptosystem, 3DES, Random Key Generator, Block Cipher Chaining, Symmetric Techniques.**

I. INTRODUCTION

Cloud computing is an emerging computing technology that uses the internet and central remote servers to maintain data and applications. Basically cloud computing is the idea of accessing files, software and computing services through the Internet instead of on our personal computer. The primary benefits of Cloud Computing are the ability to create, update and store your files through any computer that has access to the Web. It has the ability to rent a virtual server, load software on it, turn cloud services on and off at will, or clone it ten times to meet a sudden workload demands. It can be storing and securing immense amounts of data that is only accessible by authorized applications and user.

Cloud computing has the ability to use applications on the Internet that store and protect data while providing a service.

Cloud Computing moves the application software and databases to the large data centers, where the management of the data and services may not be fully trustworthy. This unique attribute, however, poses many new security challenges which have not been well understood. In this project, we focus on cloud data storage, encryption of data as well as security of the data, which has always been an important aspect of quality of service.

II. RELATED WORK

Juels et al. [1] described a formal "proof of retrievability" (POR) model for ensuring the remote data integrity. Their scheme combines spot-checking and error-correcting code to ensure both possession and retrievability of files on archive service systems. Shacham et al. [2] built on this model and constructed a random linear function based homomorphic authenticator which enables unlimited number of queries and requires less communication overhead. Bowers et al. [3] proposed an improved framework for POR protocols that generalizes both Juels and Shacham's work. Later in their subsequent work, Bowers et al. [4] extended POR model to distributed systems.

However, all these schemes are focusing on static data. The effectiveness of their schemes rests primarily on the preprocessing steps that the user conducts before outsourcing the data file F. Any change to the contents of Feven few bits, must propagate through the error-correcting code, thus introducing significant computation and communication complexity. Ateniese et al. [5] defined the "provable data possession" (PDP) model for ensuring possession of file on untrusted storages. Their scheme utilized public key based homo-morphic tags for auditing the data file, thus providing public verifiability. However, their scheme requires sufficient computation overhead that can be expensive for an entire file. In their subsequent work, Ateniese et al. [6] described a PDP scheme that uses only symmetric key cryptography.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 2, Issue 4, April 2014)

This method has lower-overhead than their previous scheme and allows for block updates, deletions and appends to the stored file, which has also been supported in our work. However, their scheme focuses on single server scenario and does not address small data corruptions, leaving both the distributed scenario and data error recovery issue unexplored. Curtmola et al. [7] aimed to ensure data possession of multiple replicas across the distributed storage system. They extended the PDP scheme to cover multiple replicas without encoding each replica separately, providing guarantees that multiple copies of data are actually maintained. In other related work, Lillibridge et al.[8] presented a P2P backup scheme in which blocks of a data file are dispersed across m+k peers using an (m+k,m)-erasure code. Peers can request random blocks from their backup peers and verify the integrity using separate keyed cryptographic hashes attached on each block. Their scheme can detect data loss from free riding peers, but does not ensure all data is unchanged. Filhoet al. [9] proposed to verify data integrity using RSA-based hash to demonstrate uncheatable data possession in peer-to-peer file sharing networks. However, their proposal requires exponentiation over the entire data file, which is clearly impractical for the server whenever the file is large. Shah et al. [10] proposed allowing a TPA to keep online storage honest by first encrypting the data then sending a number of pre-computed symmetric-keyed hashes over the encrypted data to the auditor. However, their scheme only works for encrypted files and auditors must maintain long-term state. Schwarz et al.[11] proposed to ensure file integrity across multiple distributed servers, using erasure-coding and block-level file integrity checks. However, their schemes only considers static data files and do not explicitly study the problem of data error localization

III. Proposed Methodology

Proposed trusted computing system model is categorized in two parts. First part describes the functions that can be performed on cloud for storage and management of files and the other part describes the security algorithm applied on the cloud to make it secure. Cryptography is the practice and study of techniques for secure communication in the presence of third parties.

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances

A. Hardware and Software Requirements for Development

1. *Minimum 1Gz dual core AMD BRAZO CPU or Equivalent.*
2. *Minimum 50 GB storage space on Hard Disk for User Storage.*
3. *4 GB DDR3 RAM or Above*
4. *Integrated Network Adapter*
5. *High Speed LAN Cable*
6. *Windows 7 Professional or Above*
7. *Visual Studio 2010*
8. *MySQL v5.5.3*
9. *ODBC Connector v5.5.3*
10. *Heidi SQL*
11. *IIS Management Server*
12. *Microsoft Dot-Net Framework v4.0 or above*

B. System Plan

We aim to develop a cloud storage system where a user would first have to register him on the cloud by creating an account. We have made use of symmetric key encryption for data, and for authentication of the user we make use of the technique of 2-step verification which will contain Graphical password in its second step. We use the front-end as C# language to create the application while we use MySQL as back-end technique to handle the technique of data storage. Below given is our system plan representation.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 2, Issue 4, April 2014)

TABLE 1
SYSTEM PROPOSED PLAN

SR. No.	Activity	Duration	Description
1	Project Need & Motivational Study	01 Month	To study the need of project and establishing requirements
2.	Data Analysis concerning to the current project work	02 Months	To analyze system, their disadvantages, scope for new technologies to be implemented and to study feasibility factor.
3.	Designing Modules of the Project	03 Months	To develop system architecture, different modules and to study domain of each individual module.
4	Coding and Simulations of the Project	04 Months	To generate 3DES algorithm according to need, to test connectivity for email generation and to perform various simulations for different conditions.
5.	Testing & Debugging of the Project	1 Months	To analyze each module execution and to compare output results with available system
6.	Deployment & Documentation	1 month	To look for scope of encrypting images, audio, video file to be incorporated in current execution.

IV. RESULT

We have designed the prototype model which has the desired features which we had earlier mentioned in our objectives of 2 step verification process of using a alphanumerical password in the initial stage and graphical password in the next stage for the authentication of the user. Due to the time constraints we have managed to implement encryption algorithm on textual content only. However it is feasible to encrypt images, audio video in future.

The designed prototype will be subjected to three well know globally used penetration testing tools *viz.* KBG Key-Logger, Dark Comet-RAT, High Ion Orbit Canon.

The analysis will be consisting of series of test to verify the security level to which the password can be obtained using tools. We can say that the project appears as per planned however the prototype designed gives the glimpse of the implementation of our objectives. The two step verification process provides an added layer of authentication of the user by asking him to perform two task of writing password and choosing appropriate images. Using 3DES algorithm we have successfully encrypted the .txt files while other multimedia files are safeguarded by key generation policy. Also the generated key being sent on the users email id only registered users of that email id can access the key.

1) KBG Key logger Output:

Given below are the screenshots of the KGB Key logger software which was tested on our project. As shown in the screen shot the first level of password which consists of alphanumeric characters is captured by the software however, it failed to capture the next level of authentication which is graphical password. This result ensured that the KBG Key logger was unable to crack the authentication procedure of our project completely.

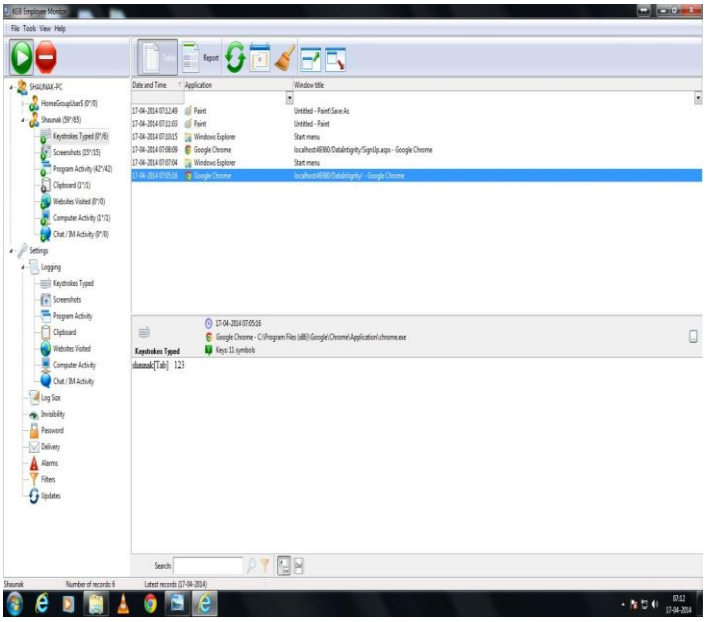

Fig. Snapshot of the software tested KGB Key logger

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 2, Issue 4, April 2014)

2) 4.7.2 Dark Comet (R.A.T)

To verify that the KBG Key logger output were verified we made use of another famous Trojan Generation tool known as the Dark Comet which is the Remote Administration Tool (R.A.T). The Key logger failed to log the graphical password of the project.

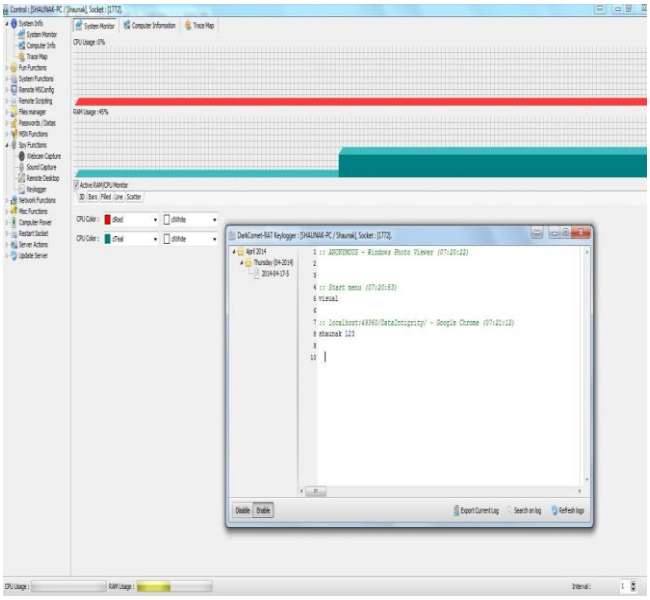

Fig. Snapshot of the software tested Dark Comet Key logger

However our project failed to provide security if a client has already been affected with a Trojan which has the feature of remote viewing the desktop activity. In such a scenario the graphical password can be viewed of the section of the password made.

V. CONCLUSION

The developed project achieved the goals of authentication and providing security on cloud storage since during our analysis the tools failed to identify the graphical password thus failing to proceed further in our project.

Also we achieved data encryption during the storing of the textual data files. This ensured that the content to be safe and secure during a scenario of breach-of-data.

This concludes us in saying that the goals that were set during the development of the project have been achieved as desired and the project is ready for large scale implementation or for commercialization.

REFERENCES

[1] A. Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. of CCS '07, pp. 584–597, 2007.

[2] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. of Asiacrypt '08, Dec. 2008.

[3] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and implementation," Cryptology ePrint Archive, Report 2008/175, 2008, http://eprint.iacr.org/.

[4] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. Of CCS '07, pp. 598–609, 2007

[6] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. of SecureComm '08, pp. 1

[7] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple- Replica Provable Data Possession," Proc. of ICDCS '08, pp. 411–420,

[8] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," Proc. of the 2003 USENIX Annual Technical Conference (General Track), pp. 29–41, 2003.

[9] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating Data Possession and Uncheatable Data Transfer," Cryptology ePrint Archive, Report 2006/150, 2006, http://eprint.iacr.org/.

[10] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop on Hot Topics in Operating Systems (HOTOS '07), pp. 1–6, 2007.

[11] T. S. J. Schwarz and E. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc. of ICDCS '06, pp. 12–12, 2006.