Research Journal of Applied Sciences, Engineering and Technology 7(14): 2946-2953, 2014
DOI:10.19026/rjaset.7.625
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: July 7, 2013 Accepted: August 21, 2013 Published: April 12, 2014

Research Article

A Comprehensive Data Forwarding Technique under Cloud with Dynamic Notification

[1]S.V. Divya, [1]R.S. Shaji and [2]P. Venkadesh
[1]Department of IT,
[2]Department of CSE, Noorul Islam Centre for Higher Education, Kumaracoil, Tamilnadu, India

Abstract: The aim of the study is to propose an integrated encrypting, encoding, forwarding and online alert scheme with a decentralized scattered code such that a secure distributed and forward storage system is formulated. The distributed data forwarded storage system supports secure and robust data storage and retrieval and also lets a user forward his data among storage servers to another user without retrieving the data back. To build a secure storage data forwarding system with dynamic online alerts that serves multiple functions is challenging when the storage system is distributed and has no dominant authority. In this study, Multiplicative Homomorphism Encryption scheme is used, which is fully Chainable, ensured correctness and compactness of data communication in the streaming of private information over forwarded data. The main technical contribution is that the Online Alert methodology supports dynamic notification to the owner of data, when unauthorized files are rehabilitated or accessed by malicious hacker during online exchange of forward data over cloud. We analyze and Report effective forms of defense and unsolicited request between cloud servers in the integrated encrypting, encoding, forwarding and online Alert scheme without significant changes to its architecture.

Keywords: Congestion control, homomorphic encryption, online alert methodology, scattered code

INTRODUCTION

A cloud storage system, consisting of distributed network storage servers, delivers increase of data availability, cost saving and hardware free storage service in a large data center of Enterprise Cloud Storage over the public internet. Cloud computing entrusts remote services with a user's data, software and computation. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. Cloud computing extends this boundary to cover servers as well as the network infrastructure. Local computers no longer have to do all the heavy lifting when it comes to running applications. Hardware and software demands on the user's side decreases. Cloud computing is an emerging concept combining many fields of computing. The foundation of cloud computing is the delivery of services, software and processing capacity over the Internet, reducing cost, increasing storage, automating systems, decoupling of service delivery from underlying technology and providing flexibility and mobility of information.

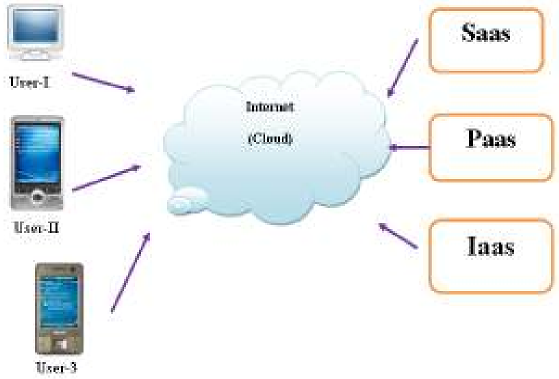


Fig. 1: Cloud computing architecture

The cloud computing architecture is shown in Fig. 1 which consists of a front end and a back end. They connect to each other through a network, usually the Internet. The front end is the side the computer user, or client, sees. The back end is the “cloud” section of the system.

The main objective of our proposed method is to provide a fully chainable, ensured correctness and compactness with integrated encrypting, encoding methodology and sending online alert notification to the holder of the data when mischievous hacker attach during data forwarding between cloud servers.

Corresponding Author: S.V. Divya, Department of IT, Noorul Islam Centre for Higher Education, Kumaracoil, Tamilnadu, India

Our contribution: Distributed networked storage systems aim to provide the storage service on the Internet. Current research on distributed networked storage systems focuses on efficiency and robustness of the storage systems. The Cloud leverage storage platform scales to support business storage needs of any size and use Virtualization technology, which automatically increases capacity to support additional customers on the fly and ensures performance is always at its best. Storing personal data, such as e-mails and photos, applications on the Internet has become a common practice but placing sensitive data in the hands of third party cloud system seem, intuitively risky when data moves or forward over public networks. We ensured accuracy and solidity of data communication in the flowing of private information over forwarded data.

BACKGROUND MATERIALS

Cong *et al.* (2006) had ensured the correctness of users' data in the cloud. This scheme achieves the data storage correctness; allow the authenticated user to access the data and data error localization, i.e., the identification of misbehaving servers. Token Pre-computation and Erasure-correcting code verification and error localization are the techniques used. It is desired that data stored in the system remain private even if all storage servers in the system are compromised. The major challenge of designing these distributed networked storage systems is to provide a better privacy guarantee while maintaining the distributed structure. To achieve this, a secure decentralized erasure code was proposed by Hsiao-Ying and Wen-Guey (2010), which combines a threshold public key encryption scheme and a variant of the decentralized erasure code which constructs a secure decentralized erasure code with low storage cost. Threshold public key encryption scheme and decentralized erasure code are the techniques used. Here, the data retrieval process may be delayed if multiple clients request is under verification.

Storing data in a third party cloud system cause serious concern over data confidentiality. Users just use services without being concerned about how computation is done and storage is managed. Hsiao-Ying and Wen-Guey (2012) had focused on designing a cloud storage system for robustness, confidentiality and functionality. General encryption schemes protect data confidentiality, but also limit the functionality of the storage system. They proposed a threshold proxy re-encryption scheme and integrate it with a decentralized erasure code. Our method fully integrates encrypting, encoding and forwarding. These parameters allow more flexible adjustment between the number of storage servers and robustness. Proxy re-encryption scheme and General encryption algorithms are used. Richard *et al.* (2005) had characterized the problems and their impact on adoption. The data in the cloud can be controlled fully using cryptographic encryption method over outsourced storage system. Security Maintenance is the major issue during data exchange since the hacker can attach or modify the data.

According to Alexandros and Vinod (2006), the problem of constructing an erasure code for storage when data storage sources are distributed is considered. For retrieving the data by querying the storage nodes, a data collector is used. Hence they introduced decentralized erasure code with Simple Randomized Network Algorithm which reduces the communication, computation and storage cost. To enable public auditability, users use a Third Party Auditor (TPA) to check the integrity of outsourced data. Cong *et al.* (2010) proposed a secure cloud storage system that supports privacy-preserving public auditing scheme to check the integrity of outsourced data. They enabled the TPA to perform auditing for multiple users simultaneously and efficiently by means of Public-Key based Homomorphic Linear authenticator technique. Data confidentiality and Data robustness are the major issues here. When data is moved into the cloud, some standard encryption methods were used to store the data and to secure the operations. Maha *et al.* (2012) had proposed an application of a method to perform and to execute operations only on the encrypted data without knowing the raw data where the calculations were actually done. In their work, they focussed on the applications of Homomorphic encryption method for cloud security using Homomorphic encryption to perform calculations on the encrypted data without decrypting them.

Bowers *et al.* (2009) had introduced a High Availability and Integrity Layer (HAIL) for cloud storage which is a distributed cryptographic system that allows many servers to prove a client that the remote file stored is retrievable and intact. On behalf of the client, a service will interact with the server for remote file integrity assurance. HAIL is based on a trusted single verifier with includes some benefits like low overhead, direct client-server communication and static file protection.

Ateniese *et al.* (2006) had demonstrated the usefulness of re-encryption scheme as a method for adding access control to secure the file system. The main advantage of their scheme is that it is unidirectional; i.e., they don't need the delegators to reveal their secret keys to anyone. In their scheme, only limited amount of trust is placed on the proxy and hence it is not able to decrypt the cipher text it encrypts before. Moreover, the keys are stored under a mater public key. When a user requests, the access control server uses proxy cryptography and re-encrypts the key to the server without the having knowledge about the key.

All the existing system mentioned above lack security in terms of data confidentiality, robustness, integrity, efficiency and performance. Moreover, they don't have the control of threshold limit and the Offline data verification cause delay in the security when hacker attacks during an online exchange of forwarding the data and cannot be forwarded between servers without retrieval of data from client machine. Apart from that, there is no secure and efficient method to forward the data from one user to another. Hence we proposed a method for efficient and secure data forwarding scheme using Multiplicative Homomorphic Encryption using Mobile alert is used.

PROPOSED METHOD AND SYSTEM ARCHITECTURE

The proposed system is shown in Fig. 2 which is used to build up a secure storage data system and provides a secure data forwarding among cloud storage servers by generating an Online Alert scheme in the client server environment under a cloud. Besides Multiplicative Homomorphic Encryption Algorithm, encrypting the data as well as limiting the external access to the cloud servers by fixing the threshold limit of the client server request and ensuring symmetric privacy and correctness during the data retrieval from large data storage family for protecting the sensitive data from internet hackers during data forwarding over Cloud storage environment is also considered.

Unlike the previous model, the proposed system consists of two servers which connect with the users. The data or information is exchanged through the servers by means of a service provider and if any unauthorized user tries to attempt to access the encrypted data, an online alert is sent to the data owner.

Data forwarding authentication: A critical element of cloud network scalability is the size of the forwarding data in network switches deployed in the data center. This factor impacts many elements of data center scalability. An entity, which has the expertise and capabilities to identify the unauthorized access and expose the risk to the source servers during data forwarding of cloud storage services.

Mobile alert: Offline data verification cause delay in the security when hacker attack during online exchange of forward data over.

An entity, which produces an alert to the cloud storage service provider or administrator who manages the cloud storage server during the unauthorized access.

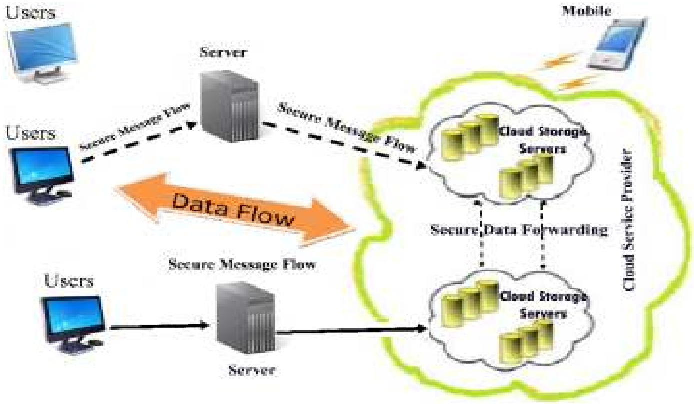


Fig. 2: Proposed system for cloud data forwarding

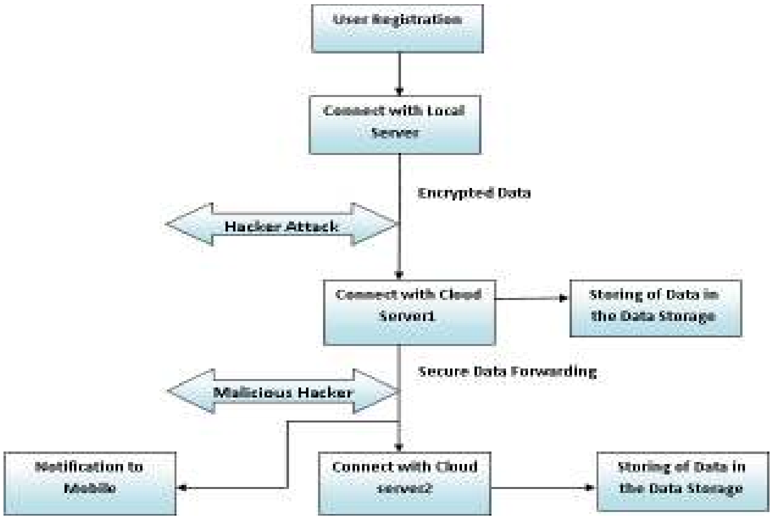

User Registration
Connect with Local Server
Encrypted Data
Hacker Attack
Connect with Cloud Server1
Storing of Data in the Data Storage
Secure Data Forwarding
Malicious Hacker
Notification to Mobile
Connect with Cloud server2
Storing of Data in the Data Storage

Fig. 3: Data flow diagram of proposed system

The proposed model is described clearly with the help of data flow diagram as shown in Fig. 3. The user who wishes to forward the data has to enter the username and password if already registered; otherwise he has to register with the server and login using username and password. If valid, the user is connected to the server and the encrypted data is forwarded securely from one cloud server to another cloud server. If the encrypted data are accessed by malicious users, the notification is sent to the owner's mobile.

Pseudo code:

```
Step 1: User connects with the local server by giving
        the username and password
Step 2: After connecting with the local server, the user
        encrypts the data and the encrypted data is
        stored in the data store of the cloud server
Step 3: If (authorized access to data)

then
forward the data to another server
else
notification is sent through mobile to data owner
```

METHODOLOGY

Multiplicative Homomorphic encryption is used. The problem of Homomorphic Encryption arises in many practical applications where privacy is required for some functionality, including cloud computing, private information retrieval, (private) search on encrypted/streaming data, etc.

Mathematical modeling: The input for a server, Alice, is some function f, whereas the input of a client, Bob, is some value x in the domain of f. This problem involves three steps:

- **Encryption:** Bob uses some randomized algorithm Enc to produce and send to Alice an encrypted version of x, $c1 \leftarrow Enc\ (pk, x)$.
- **Evaluation:** Alice applies some evaluation algorithm Eval to her input f and the value c1, to produce $c2 \leftarrow Eval\ (f, c1)$, which is sent back to Bob.
- **Decryption:** Bob uses a decryption algorithm Dec (sk, c2) to extract from c2 the reply's value y.

The paradigm works correctly if $y = f(x)$ for any value of x in the domain of f, any key pair (pk, sk) and any cipher text $c1 \in Enc\ (pk, x)$.

The Homomorphic encryption scheme is based on three criteria: the variety of functions for which correctness is ensured, the privacy obtained by them, and the compactness of the communications.

An encryption pattern permitting to compute a limited amount of sums and products over encrypted data with a security reduction. This scheme is able to assess each request with exponentially many operands and can be converted into an effectual fully-homomorphic encryption scheme

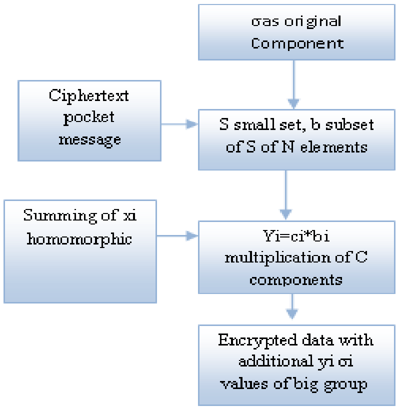

σas original Component
Ciphertext pocket message
S small set, b subset of S of N elements
Summing of xi homomorphic
Yi=ci*bi multiplication of C components
Encrypted data with additional yi σi values of big group

Fig. 4: Sender side diagram

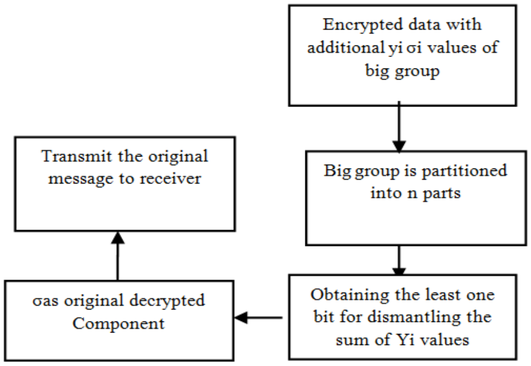

Encrypted data with additional yi σi values of big group
Transmit the original message to receiver
Big group is partitioned into n parts
σas original decrypted Component
Obtaining the least one bit for dismantling the sum of Yi values

Fig. 5: Receiver side diagram

A technique for homomorphic encryption is by providing a cipher text with component c, there exists a big group which is subset of small group and it is summing up the elements of the small group yields the private key and homomorphically summing all elements to obtain a resulting cipher text that is an encryption of the at least one bit, where the big group is subdivided into n chunks with each part having a multiplicity of different elements from the big group.

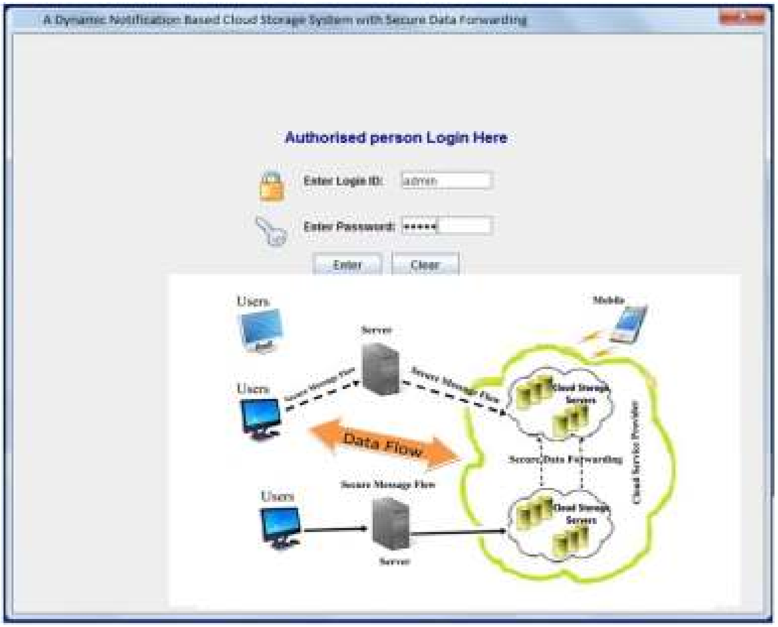
Send request: The sender side diagram is shown in Fig. 4. The requester R has sent the message as cipher steam C, it formed the set of bit vector with is added with summing of secret format of the message which if the multiplication of set big group who forms the encrypted data:

$R = \{\sigma_i, S, b € S, y_i, E\ (M)\}$; $y_i = c_i{*}b_i$ and E (M) is the Encrypted Message

Receiver message: The receiver decrypts the chipper message evaluation the yiσi; and homomorphically summing all xi as shown in Fig. 5 to obtain a resulting cipher text that is an encryption of the at least one bit, where the big group is partitioned into n parts with each part having a plurality of different elements from the big group, where the elements of the small group are one element from each part.

Implementation: The java is used for implementation. Here, we have investigated the problem of security during the data forwarding in the cloud data storage system, which is essentially a distributed storage system. To ensure the accuracy and security of users' data we proposed an effective and flexible distributed scheme with explicit dynamic data support, including block, update, delete, append, forwarding, alert and notification. We rely to provide redundancy parity vectors and guarantee the data dependability by utilizing the Multiplicative Homomorphism token with distributed verification of erasure coded data. The admin can login to the server by giving his username and password as shown in Fig. 6.

Notification of authorized user, the client login form and accessing the cloud file using the secret key and the mobile alert is shown in Fig. 7 to 10, respectively.



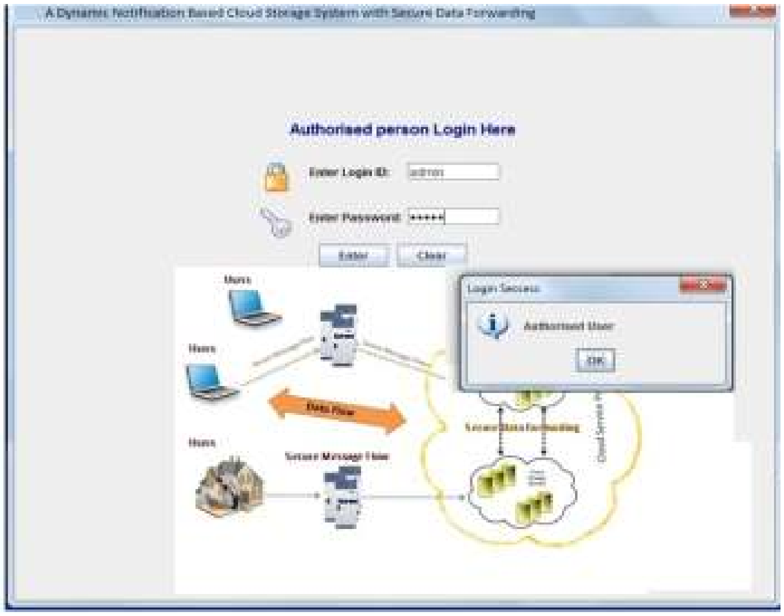
A Dynamic Notification Based Cloud Storage System with Secure Data Forwarding

Authorised person Login Here

Enter Login ID: admin

Enter Password: *****

Fig. 6: Admin login to access cloud storage server



A Dynamic Notification Based Cloud Storage System with Secure Data Forwarding

Authorised person Login Here

Enter Login ID: admin

Enter Password: *****

Login Success

Authorised User

Fig. 7: Notification of authorized user

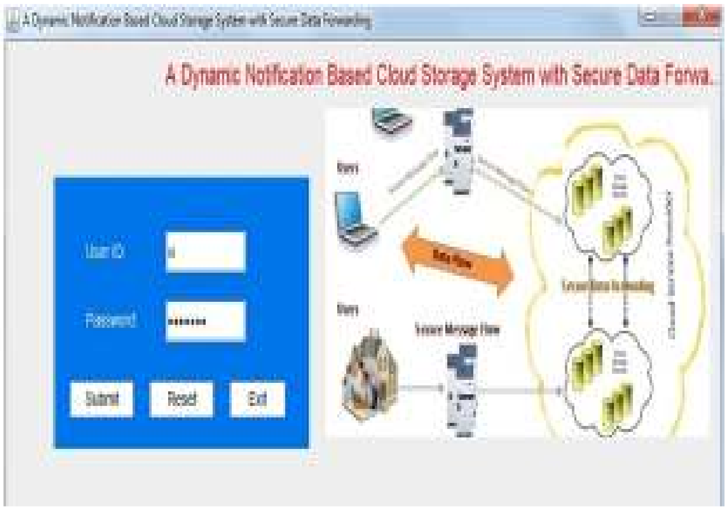

A Dynamic Notification Based Cloud Storage System with Secure Data Forwarding
A Dynamic Notification Based Cloud Storage System with Secure Data Forwa
User ID
Password
Submit
Reset
Exit
Users
Data Flow
Users
Secure Message Flow
Secure Data Forwarding
Cloud Storage Servers

Fig. 8: Client login from client machine

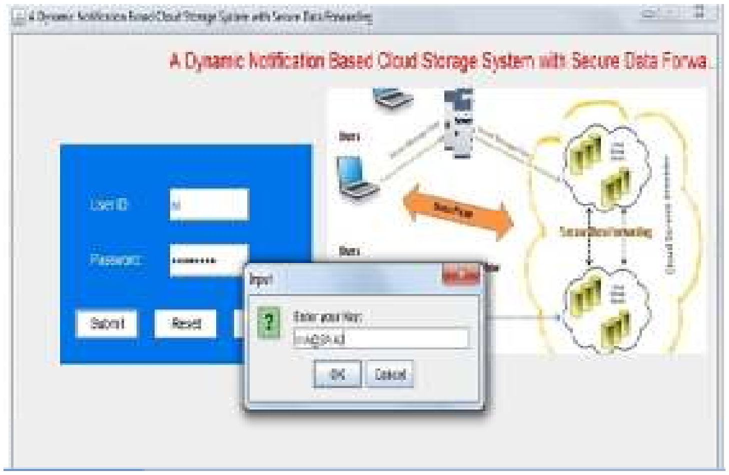
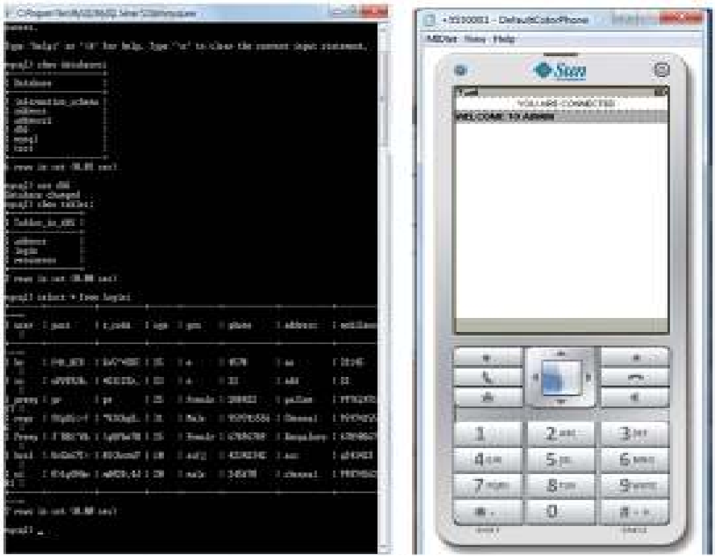

A Dynamic Notification Based Cloud Storage System with Secure Data Forwarding
A Dynamic Notification Based Cloud Storage System with Secure Data Forwa
User ID
Password
Submit
Reset
Users
Data Flow
Users
Secure Data Forwarding
Cloud Storage Servers
Input
Enter your Key:
OK
Cancel

Fig. 9: Accessing the cloud file by using secret key



```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| [illegible]        |
| [illegible]        |
| [illegible]        |
| mysql              |
| test               |
+--------------------+
6 rows in set (0.00 sec)

mysql> use [illegible]
Database changed
mysql> show tables;
+------------------+
| Tables_in_[illegible] |
+------------------+
| [illegible]      |
| login            |
| [illegible]      |
+------------------+
3 rows in set (0.00 sec)

mysql> select * from login;
[illegible]
7 rows in set (0.00 sec)

mysql> _
```

Fig. 10: Cloud storage server and mobile alert

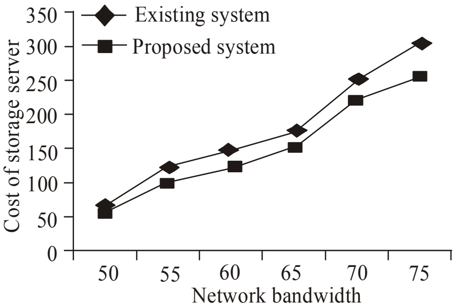


Fig. 11: Simulation graph for network bandwidth vs. storage server cost

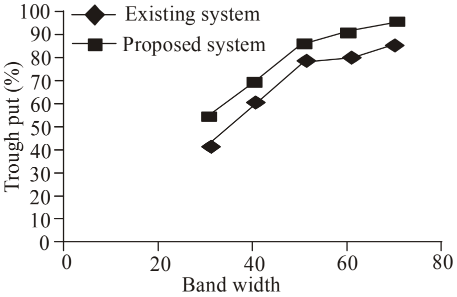


Fig. 12: Simulation graph for bandwidth vs. throughput

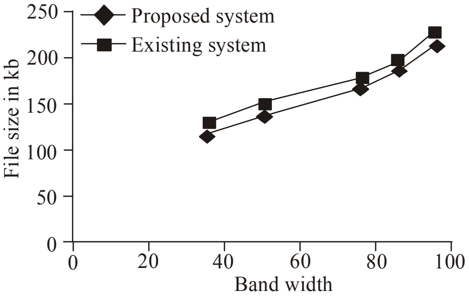


Fig. 13: Simulation graph for bandwidth vs. file size

RESULTS AND DISCUSSION

The existing system is compared with that of our proposed system in-terms of Network bandwidth and Storage server cost and shown in Fig. 11 and Table 1.

The existing system is compared with that of our proposed system in-terms of Network bandwidth and Throughput and is shown in Fig. 12 and Table 2.

The existing system is compared with that of our proposed system in-terms of Network bandwidth and Storage server cost and shown in Fig. 13 and Table 3.

Our experiments shows that the proposed system is more advantageous when compared to existing system in terms of Network Bandwidth, Throughput and File Size.

Our scheme achieves the integration of storage with secure forwarding and online alert notification to the service provider when unauthorized files are amended or accessed by malicious hackers during an online exchange of forward data over cloud which is value added. Besides, this scheme is also very beneficial for:

Table 1: Comparison chart of existing and proposed system with respect to network bandwidth and cost of storage server

Network bandwidth	Cost of storage server (existing system)	Cost of storage server (proposed system)
50	65	55
55	120	100
60	145	122
65	175	152
70	250	220
75	300	255

Table 2: Comparison chart of existing and proposed system with respect to network bandwidth and throughput

Network bandwidth	Throughput (existing system) (%)	Throughput (proposed system) (%)
30	42	55
40	60	68
50	77	85
60	79	90
70	84	94

Table 3: Comparison chart of existing and proposed system with respect to network bandwidth and throughput

Network bandwidth	File size (in KB) of existing system	File size (in KB) of proposed system
35	115	125
50	135	150
75	165	175
85	185	195
95	210	225

- Congestion control
- Optimization of the number of request between cloud servers

CONCLUSION

We can almost guarantee the simultaneous identification of the misbehaving servers, whenever data corruption or unauthorized client has been detected during the data forwarding across the distributed servers. This scheme is very expedient during number of cloud servers request optimization which leads congestion control between cloud servers.

Although, our scheme is highly efficient and resilient to scheming failure, malicious data modification attack and even server colluding attacks when compared to recent findings and researches. However malicious users can find the way to stop sending alert message to the owner of the cloud storage organization during unauthorized access of the files so end-to-end security is definitely a challenging job in the cloud storage environment. Moreover, the focus is also being done on:

- Size/Length of the encrypted message
- No. of packets to be transmitted

REFERENCES

Alexandros, G.D. and P. Vinod, 2006. Decentralized erasure codes for distributed networked storage. IEEE Trans., 52(6): 2809-2816.

Ateniese, G., K. Fu, M. Green and S. Hohenberger, 2006. Improved proxy re-encryption schemes with applications to secure distributed storage. ACM T. Inform. Syst. Secur., 9(1): 1-30.

Bowers, K.D., A. Juels and A. Oprea, 2009. HAIL: A high-availability and integrity layer for cloud storage. Proceeding of the 16th ACM Conference on Computer and Communication Security (CCS), pp: 187-198.

Cong, W., W. Qian and R. Kui, 2006. Ensuring Data Storage Security in Cloud Computing. pp: 63-73. Retrieved from: eprint.iacr.org/2009/081.pdf.

Cong, W., S.M.C. Sherman and W. Qian, 2010. Privacy-preserving public auditing for secure cloud storage. Proceeding of the IEEE Conference, pp: 525-533.

Hsiao-Ying, L. and T. Wen-Guey, 2010. A secure decentralized erasure code for distributed networked storage. IEEE Trans., 21(11): 1586-1594.

Hsiao-Ying, L. and T. Wen-Guey, 2012. A secure erasure code based cloud storage system with secure data forwarding. IEEE Trans., 23(6).

Maha, T., E.H. Saïd and E.G. Abdellatif, 2012. Homomorphic encryption applied to cloud computing security. Proceedings of the World Congress on Engineering (WCE 2012), July 4-6, Vol. 1.

Richard, C., G. Philippe and M. Jesus, 2005. Controlling data in the cloud: Outsourcing computation without outsourcing control. Proceedings of the IEEE Conference, pp: 136.