A NOVEL APPROACH FOR DATA STORAGE SECURITY IN A HOSTED ENVIRONMENT

Srinivas Jagirdar[1], K.Venkata SubbaReddy[2], Dr Ahmed Abdul Moiz Qyser[3], Dr.K.Uday Kumar[4]

Department of Computer Science and Engineering
Muffakham Jah College of Engineering and Technology
Banjarahills, Hyderabad, Andhra Pradesh, India.
jagirdar.srinivas@gmail.com[1] kvsreddy2012@gmail.com[2]
aamoiz@gmail.com[3] udaikudikyala@gmail.com[4]

ABSTRACT

The introduction of hosted environments to computing has brought a lot of change in the industry. With the advent of Infrastructure-as-a-Service, the concept of renting data stores is catching up. Security for the data in a hosted environment is an alarming issue. The paper focuses on cloud data storage security. We propose a security model to maintain the data store. This model ensures the safety of data in the hosted environment. By utilizing the homomorphic token with random masking the model achieves the abstraction of data stored on the Hosted Environment.

KEYWORDS:

Data Store, Security, Hosted Environment, Infrastructure-as-a-Service.

1. INTRODUCTION

Cloud computing is a model for enabling ubiquitous, 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 interaction [1]. Cloud computing not only reduces the cost of service delivery but also increases the speed and agility with which services are deployed. Cloud computing incorporates virtualization, on-demand deployment, Internet delivery of services, and open source software [2].The characteristics of cloud computing include On-demand self-service, Broad network access, Resource pooling, Rapid elasticity and Measured service. Even though cloud offers a lot of advantages there are some issues that cloud has to deal with. Security in cloud computing is a major concern in the industry. Many companies are still waiting for an amicable solution for this major problem to be solved as they don't want to take any risks with their critical & sensitive data. The unique issues associated with cloud computing security have not been resolved yet. Access to your information from anywhere at any time is the specialty of hosted environments. You don't need to be in the same physical location as the hardware that stores your data. The Cloud provider houses the hardware and software necessary to run your applications.

DOI : 10.5121/ijcsea.2013.3104

Cloud computing raises a lot of security threats. Traditional cryptographic primitives can not be directly adopted for data security in a hosted environment because the user losses control over the data. Verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. The problem of verifying correctness of data storage in the cloud is a challenge because various kinds of data for each user is stored in the cloud and also there is a demand for long term continuous assurance of this data. Data storage in a Hosted environment is not just a third party data warehouse. The data stored in the Hosted environment may be frequently updated by the users. The various operations include insertion, deletion, modification, appending, reordering, etc. Ensuring storage correctness under dynamic data update is a significant task. Traditional integrity insurance techniques don't work here therefore new solutions are required. The deployment of data Hosted environment is powered by data centers running in a simultaneous, cooperated and distributed manner. In order to reduce the data integrity threats, data is redundantly stored in multiple physical locations. Therefore, distributed protocols for storage correctness, assurance can be employed for cloud data storage system. But the field is still evolving. Recent research is revolving around the importance of ensuring the remote data integrity .The techniques discussed in [3]-[7], can be useful to ensure the storage correctness without having users possessing data, can not address all the security threats in cloud data storage, since they are all focusing on single server scenario and most of them do not consider dynamic data operations. None of the proposed distributed schemes can be applied to dynamic data scenario. As a result, their applicability in cloud data storage can be drastically limited. In this paper, to achieve the abstraction of data stored on a Hosted Environment, we propose a security model that utilizes the homomorphic token with random masking technique. The rest of the paper is organized as follows: Section 2 throws light on basic architecture of a data store in a hosted environment. Section 3 gives insight of Cloud Data Storage using Homomorphic Authenticator. Section 4 deals with the results & analysis part. Section 5.6 talk about the future works & Conclusions respectively.

2. BASIC ARCHITECTURE

2.1. System Model

Figure 1.illustrates the representative network architecture for cloud data storage. Three different network entities exist:

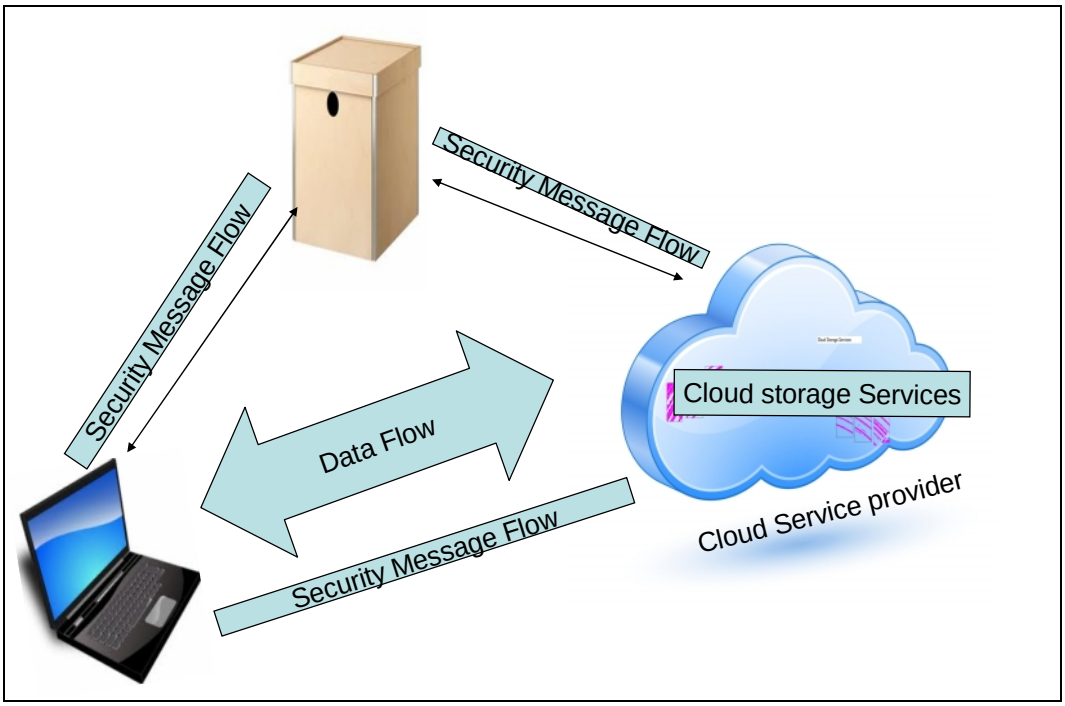
Clients: clients comprise of users who have data to be stored in the cloud. Clients rely on the cloud for data computation. They include individual consumers and organizations.

Cloud Service Provider (CSP): A CSP has resources and expertise in managing and building distributed cloud storage servers. A CSP owns and operates live Cloud Computing systems.

Third Party Auditor (TPA): An optional TPA is employed to assess and expose risk of cloud storage services on behalf of the clients. A client stores his data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Faults or server crashes can be tolerated by employing Data redundancy with technique of erasure-correcting code Thereafter, for application purposes, the user interacts with the cloud servers via CSP to access or retrieve his data. The basic operations that a client performs include block update, delete, insert and append. As Client no longer possesses their data locally, it is of critical importance to assure users that their data are being correctly stored and maintained. The clients must be equipped with

security means to make continuous correctness assurance of their stored data even if the client doesn't posses a local copy. A TPA on behalf of the client can monitor the data, if the client doesn't posses the resources, time.

Fig. 1: Hosted Environment Data Storage Architecture



3. CLOUD DATA STORAGE USING HOMOMORPHIC AUTHENTICATOR

In cloud data storage system, the data is stored in the cloud and the client no longer has access to the data. The availability & correctness of the data being stored in the Hosted Environment must be assured. Effective detection of any unconstitutional data variation and corruption is the key issue. Hence we propose this model.

The proposed model contains four modules:-

1. **Key generation module: -** In this module we use a key generation algorithm. User runs this module and sets up the scheme.

2. **Signature generation module: -** In this module we use a signature generation algorithm. Verification metadata is generated by the client through this module. Metadata may consist of MAC, signatures or other information used for auditing.

3. **Proof Generation module**: - In this module we use a Proof generation algorithm. Cloud server generates a proof of data storage correctness by running this module.

4. **Proof Verification Module:** - In this module we use a Proof Verification algorithm. TPA audits the proof from the cloud server by running this module.

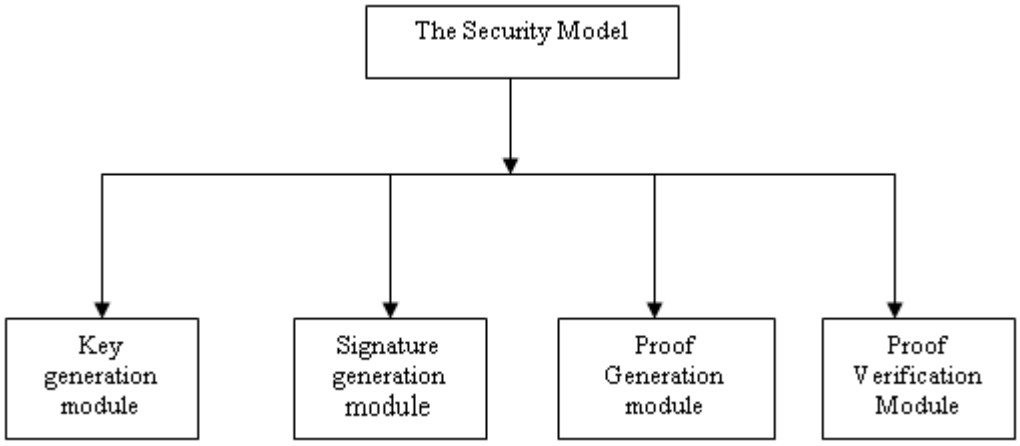


Fig2. The Security Framework

The following are the steps used in the framework:-

1. Public key & secret key generation.
2. File blocks code generation.
3. Blocks Migration in to cloud.
4. Challenge message generation.
5. Cloud Service Provider authentication.
6. Verification.

The first three steps are termed as the setup phase & the last three are termed as the audit phase.

1. Public key & secret key generation: - The user generates public and secret parameters. Key generation algorithm is used.

2. File blocks code generation:- A code is generated by the user for each file block using homomorphic authenticator. We also use a random mask achieved by a Pseudo Random Function (PRF). By looking only at the aggregated authenticator, a linear combination of data blocks can be checked.

$$\mu' = \sum_{i \in I} \nu_i m_i$$

Where vi are random number, mi are file blocks.

If TPA finds several linear combinations of the similar blocks, it might be able to infer the file blocks. A random mask given by the Pseudo Random Function (PRF) is used.

$$\mu = r + \gamma\mu'$$

Where r is the mask.

Fig 3. Depicts the generation of file blocks using homomorphic authenticator.

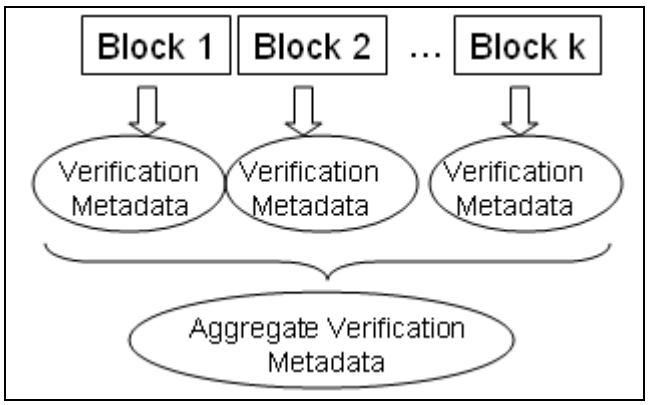


Fig.3 Homomorphic authenticator verifying file blocks

The PRF function masks the data. Verification Metadata is not affected by PRF. The Blocks without PRF mask & with PRF mask are verified. If both of them are equal then only they are authenticated by the aggregate authenticator. Figure 4 depicts this.

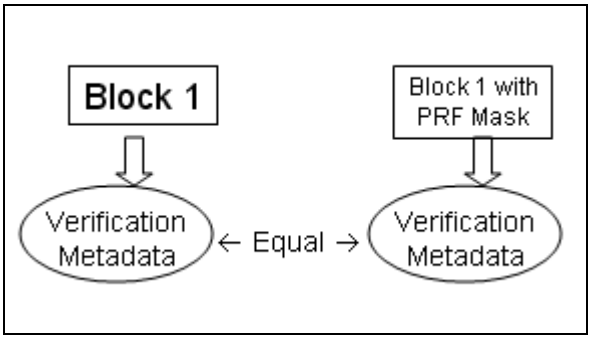


Fig.4 Random Mask by PRF

3. Blocks Migration in to cloud**:** - The codes, file blocks are migrated to the cloud.

Figure 5 depicts the setup phase.

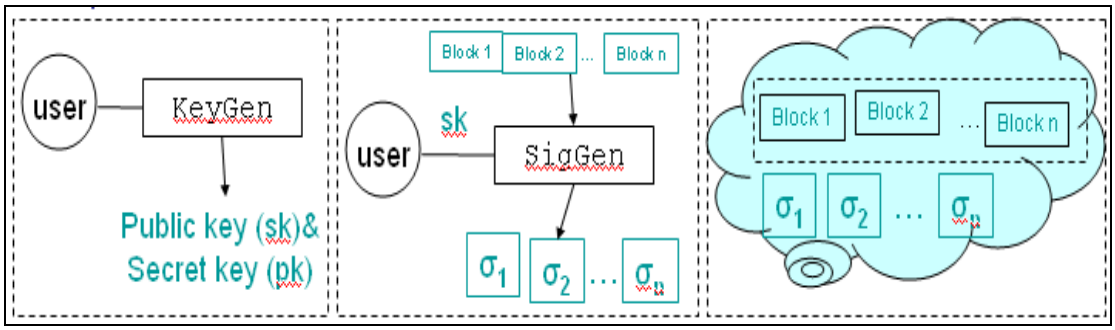


Fig.5 Setup phase

4. Challenge Message Generation: - The third party auditor sends a challenge message to the Cloud Service Provider. The position of the blocks is present in the challenge message. These positions will be checked in this phase.

5. Cloud Service Provider authentication: - A proof generation algorithm is run by the Cloud server. It generates a proof of data storage correctness. The CSP picks the file blocks generated in the challenge, applies the Proof Generation algorithm and generates the proof. The selected blocks are linearly combined by the CSP. These blocks are applied a mask. Separate PRF key is used for each audit. The CSP sends aggregate authenticator & masked combination of the blocks to TPA for further processing.

6. Verification: - In this step the proof is verified by the Verification algorithm. The Third Party Authenticator generates an aggregate authenticator. It verifies aggregate authenticator & masked combination of blocks received from the Cloud Service Provider by comparing it with the obtained Aggregate authenticator. Figure 6 depicts it.

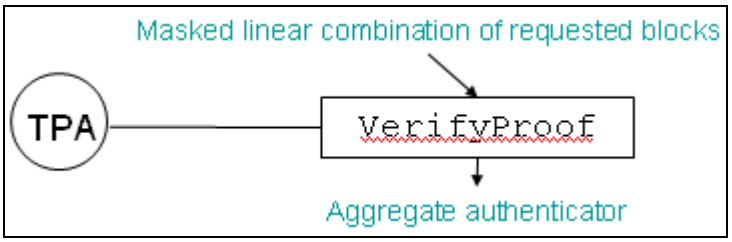


Fig 6 TPA Verification

```
Algorithm keyGen (p,s)
{
  //user generates the public, secret keys.
  Input public parameters p;
  Input private parameters s;
  Generate Public key p(k);
  Generate Secret key s(k);
  End;
}
```

```
Algorithm SigGen (m1,m2,m3…mn)
{
 //client generates aggregated authenticator for the file blocks.
 Use homomorphic authenticator.
 Use random mask.
 Generate σ1, σ2, σ3……σn  for m1,m2,m3…mn and aggregate them.
 Send it to CSP with respective file blocks.
}
```

```
Algorithm Genproof (TPAchallenge blocks)
{
//CSP generates aggregated authenticator
Receive challenge blocks from TPA.
Combine linearly the blocks.
Apply mask on the received blocks
Generate aggregate authenticator.
Send the masked combination of blocks + aggregate authenticator to TPA for verification
End
}
```

```
Algorithm Verifyproof(aggregate authenticator, masked combination of blocks)
{
//TPA verifies the received aggregate authenticator and generated authenticator.
Receive CSP computed aggregate authenticator.
If Received aggregated authenticator == generated aggregated authenticator
Return File block secure
Else
File block tampered.
}
```

4. ANALYSIS & RESULTS:

Table 1. Performance analysis of the proposed model.

Factors	Our Model		Public Auditing Model	
Selected Blocks	380	300	380	300
Server Computing Time(ms)	339.52	270.20	407.66	265.87
TPA Computing Time(ms)	419.47	476.81	504.25	472.55
Cost per Byte	132	40	132	40

The Table 1 compares our model with the existing Public Auditing Model. We need 300 or 380 blocks to detect that with a probability larger than 95% or 99%, respectively if the server is missing 1% of the data. The data transmitted from CSP to TPA is independent of the data size and the Linear combination with mask.

5. FUTURE WORK

We can extend our model into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Even the dynamics of the data on the cloud can be modified so as to adapt to any type of application.

6. CONCLUSION

In this paper, we investigated the problem of security of data storage in a hosted environment. We have proposed an effective model that ensures the correctness of clients' data in hosted environment. By utilizing the Homomorphic token with Random masking, our model achieves the abstraction of the data stored on the cloud. Our model eliminates the burden of client from the tedious and possibly expensive auditing task. It alleviates the clients' fear of their outsourced data leakage. Through detailed security and performance analysis, we show that our model is highly efficient and robust. It prevents any malicious data modification attack, and even server colluding attacks.

REFERENCES

[1] PeterMell,Timothy and Grance "The NIST Definition of Cloud Computing" http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[2] White Paper Presents General Concepts, Architectual Models & Considerations June 29, 2009, Volume 137, Issue 1

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

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

[5] 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/.

[6] 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.

[7] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. of SecureComm '08, pp. 1– 10, 2008.

[8] 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.

[9] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing." 2009.

[10] A. Ruiz-Alvarez and M. Humphrey, "An Automated Approach to Cloud Storage Service Selection," in 2nd Workshop on Scientific Cloud Computing (Science Cloud 2011), 2011.

[11] M. Berkelaar, K. Eikland, and P. Notebaert, "lpsolve: Open source (mixed-integer) linear programming system," 2011. [Online]. Available: http://lpsolve.sourceforge.net/.

[12] W. W. Chu, "Optimal File Allocation in a Multiple Computer System," IEEE Transactions on Computers, vol. 18, no. 10, pp. 885- 889, Oct. 1969.

[13] R. G. Casey, "Allocation of copies of a file in an information network," In Proceedings of the AFIPS Joint Computer Conferences, 1972, pp. 617-625.

[14] E. Grapa and G. G. Belford, "Some theorems to aid in solving the file allocation problem," Communications of the ACM, vol. 20, no. 11, p. 878, 1977.

[15] K. Lam and C. T. Yu, "An approximation algorithm for a file allocation problem in a hierarchical distributed system," in In Proceedings of the 1980 ACM SIGMOD International Conference on Management of Data, 1980, pp. 125 - 132.

[16] S. Mahmoud and J. S. Riordon, "Optimal allocation of resources in distributed information networks," ACM Transactions on Database Systems (TODS), vol. 1, no. 1, p. 66, 1976.

[17] L. W. Dowdy and D. V. Foster, "Comparative Models of the File Assignment Problem," ACM Computing Surveys (CSUR), vol. 14, no. 2, p. 287, 1982.

AUTHORS' PROFILE:

Mr.J.Srinivas obtained his Bachelor's degree in computers science & Information Technology from JNTU, Hyderabad in 2005 and received the Masters Degree in Software Engineering from JNTUH, in 2010. He is currently pursuing Ph.D.,in Computer Science and Engineering, at JNTUA, Andhra Pradesh, India His research interests include Cloud Computing, Currently he is working as Assistant Professor in Computer Science & Engineering Department at Muffakham Jah College of Engineering & Technology, Banjarahills, Hyderabad.

Mr.K.Venkata Subba Reddy obtained his Bachelor's degree in Information Technology from University of Madras in 2002 and received the Masters Degree in Software Engineering from Bharath University, Chennai in 2005, He is currently pursuing Ph.D., in Computer Science and Engineering, at Acharya Nagarjuna University, Andhra Pradesh, India. His research interests include Software Engineering, Web Technologies, Cloud computing. He is a life member of ISTE and a member of CSI. Currently he is working as Assistant Professor in Computer Science & Engineering Department at Muffakham Jah College of Engineering & Technology, Banjarahills, Hyderabad.

Dr. Ahmed Abdul Moiz Qyser received his B.E. (CSE) from Osmania University, M.Tech. (Software Engineering) from JNTU, Hyderabad, and Ph.D. from Osmania University. His research focus is Software Process Models and Metrics for SMEs. He is presently working as Professor and Head in Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Hyderabad, India. He is also a visiting Professor to the industry where he teaches Software Engineering and its related areas. He is the author of several research papers in the area of Software Engineering. He is an active member of ACM, CSI and ISTE

Dr.K.Udai Kumar, He has completed M.S and Ph.D for Mississippi State University. He is a member of IEEE.He is working as Associate Professor in the department of CSE at Muffakham Jah College of Engineering and Technolology, Banjarahills, Hyderabad.