ENABLING CLOUD STORAGE AUDITINGVERIFYING WITH VALIDREDISTRIBUTE KEY CHANGE

Are Dinesh Kumar [1], Prof. Dr. K. Babu Rao[2]

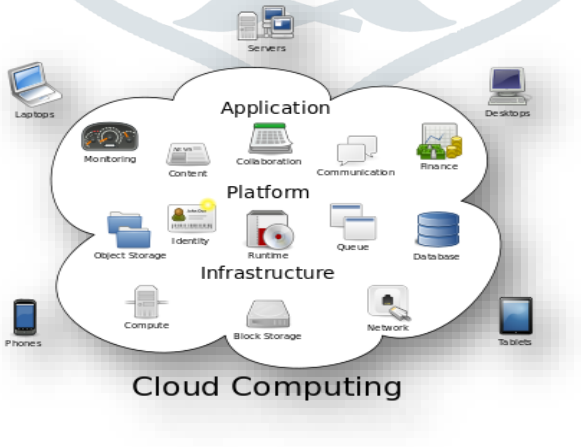
1. **M.Tech Scholar, Department of CSE, Vaageswari College of Engineering, Karimnagar, Telangana, India– aredinesh@gmail.com, 8977047764**
2. **Professor, Department of CSE , Vaageswari College of Engineering, Karimnagar, Telangana, India-s4prinicipal@gmail.com, 9502588608**

ABSTRACT:

Key-exposure resistance has always been animportant issue for in-depth cyber defence in many securityapplications. Recently, how to deal with the key exposure problemin the settings of cloud storage auditing has been proposed andstudied. To address the challenge, existing solutions all requirethe client to update his secret keys in every time period, whichmay inevitably bring in new local burdens to the client, especiallythose with limited computation resources, such as mobile phones.In this paper, we focus on how to make the key updates astransparent as possible for the client and propose a new paradigmcalled cloud storage auditing with verifiable outsourcing of keyupdates. In this paradigm, key updates can be safely outsourcedto some authorized party, and thus the key-update burden on theclient will be kept minimal. In particular, we leverage the thirdparty auditor (TPA) in many existing public auditing designs, letit play the role of authorized party in our case, and make it incharge of both the storage auditing and the secure key updates forkey-exposure resistance. In our design, TPA only needs to hold anencrypted version of the client's secret key while doing all theseburdensome tasks on behalf of the client. The client only needs todownload the encrypted secret key from the TPA when uploadingnew files to cloud. Besides, our design also equips the client withcapability to further verify the validity of the encrypted secretkeys provided by the TPA. All these salient features are carefullydesigned to make the whole auditing procedure with key exposureresistance as transparent as possible for the client. We formalizethe definition and the security model of this paradigm. Thesecurity proof and the performance simulation show that ourdetailed design instantiations are secure and efficient.

I.INTRODUCTION

What is cloud computing?



Cloud computing resources are delivered as a service Internet). The name of a cloud-shaped the complex system diagrams. Cloud services with a user's computation. Cloud hardware and software the Internet as managed computing is the use of (hardware and software) that over a network (typically the comes from the common use symbol as an abstraction for infrastructure it contains in computing entrusts remote data, software and computing consists of resources made available on third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

Structure of cloud computing

How Cloud Computing Works?

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

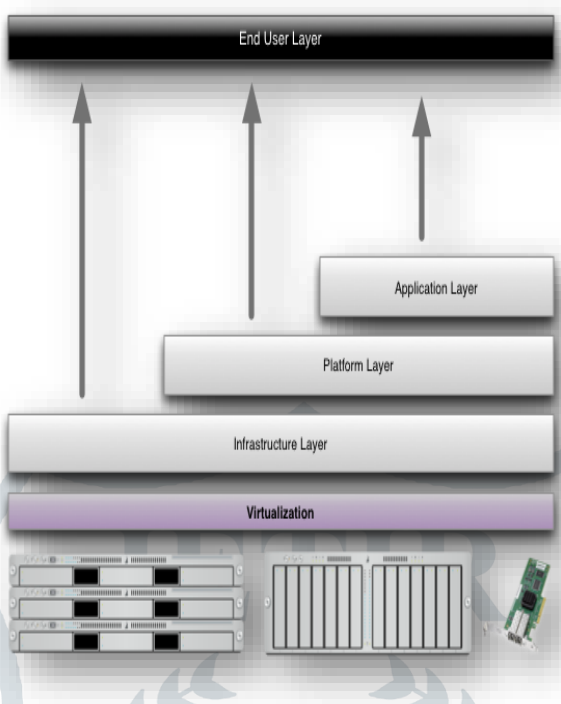
Characteristics and Services Models:

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

- **On-demand self-service**: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- **Broad network access**: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- **Resource pooling**: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.
- **Rapid elasticity**: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- **Measured service**: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

Services Models:

Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.



Structure of service models

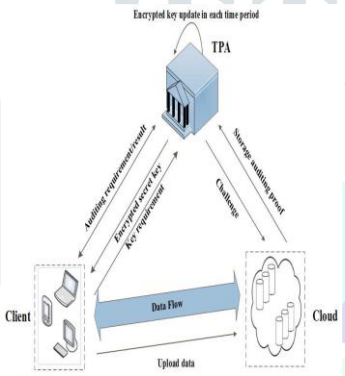
Benefits of cloud computing:

1. **Achieve economies of scale** – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.
2. **Reduce spending on technology infrastructure.** Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
3. **Globalize your workforce on the cheap.** People worldwide can access the cloud, provided they have an Internet connection.
4. **Streamline processes.** Get more work done in less time with less people.
5. **Reduce capital costs.** There's no need to spend big money on hardware, software or licensing fees.
6. **Improve accessibility.** You have access anytime, anywhere, making your life so much easier!
7. **Monitor projects more effectively.** Stay within budget and ahead of completion cycle times.
8. **Less personnel training is needed.** It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.
9. **Minimize licensing new software.** Stretch and grow without the need to buy expensive software licenses or programs.
10. **Improve flexibility.** You can change direction without serious "people" or "financial" issues at stake.

Advantages:

1. **Price:** Pay for only the resources used.
2. **Security**: Cloud instances are isolated in the network from other instances for improved security.
3. **Performance:** Instances can be added instantly for improved performance. Clients have access to the total resources of the Cloud's core hardware.
4. **Scalability:** Auto-deploy cloud instances when needed.
5. **Uptime:** Uses multiple servers for maximum redundancies. In case of server failure, instances can be automatically created on another server.
6. **Control:** Able to login from any location. Server snapshot and a software library lets you deploy custom instances.
7. **Traffic:** Deals with spike in traffic with quick deployment of additional instances to handle the load.

II.SYSTEM ARCHITECTURE:



III.EXISTING SYSTEM:

- Yu *et al.* constructed acloud storage auditing protocol with key-exposure resilienceby updating the user's secret keys periodically. In this way,the damage of key exposure in cloud storage auditing can bereduced. But it also brings in new local burdens for the clientbecause the client has to execute the key update algorithm ineach time period to make his secret key move forward.
- Forsome clients with limited computation resources, they mightnot like doing such extra computations by themselves in eachtime period. It would be obviously more attractive to makekey updates as transparent as possible for the client, especiallyin frequent key update scenarios.
- Wang *et al*. proposed a public privacy-preserving auditing protocol. They used the random masking technique to make the protocol achieve privacypreserving property.

Disadvantages of Existing System:

- Existing system don't like auditing protocol with verifiable outsourcing of key updates.
- Third party has the access to see client's secret key without encryption.
- No verification system available for client's for to check validity of the encrypted secret keys when downloading them from the TPA
- All existing auditing protocols are all built on the assumption that the secret key of the client is absolutely secure and would not be exposed.

IV.PROPOSED SYSTEM:

The main contributions are as follows:

(1) We propose a new paradigm called cloud storageauditing with verifiable outsourcing of key updates. In thisnew paradigm, key-update operations are not performed bythe client, but by an authorized party. The authorized partyholds an encrypted secret key of the client for cloud storageauditing and updates it under the encrypted state in each timeperiod. The client downloads the encrypted secret key fromthe authorized party and decrypts it only when he would liketo upload new files to cloud. In addition, the client can verifythe validity of the encrypted secret key.

(2) We design the first cloud storage auditing protocol withverifiable outsourcing of key updates. In our design, the thirdpartyauditor (TPA) plays the role of the authorized party whois in charge of key updates.

(3) We formalize the definition and the security model of the cloud storage auditing protocol with verifiable outsourcing of key updates. We also prove the security of our protocol in the formalized security model and justify its performance by concrete implementation.

Advantages of Proposed System:

- The TPA does not know the real secret key of the clientfor cloud storage auditing, but only holds an encrypted version.In the detailed protocol, we use the blinding technique withhomomorphic property to form the encryption algorithm toencrypt the secret keys held by the TPA. It makes our protocolsecure and the decryption operation efficient.
- Meanwhile, theTPA can complete key updates under the encrypted state. Theclient can verify the validity of the encrypted secret key whenhe retrieves it from the TPA.
- The client downloads the encrypted secret key from the authorized party and decrypts it only when he would like to upload new files to cloud. In addition, the client can verify the validity of the encrypted secret key.
- Cloud storage auditing protocol with verifiable outsourcing of key updates
- The client can verify the validity of the encrypted secret key when he retrieves it from the TPA
- The security model of the cloud storage auditing protocol with verifiable outsourcing of key updates.

V.IMPLEMENTATION

MODULES

- Client Module.
- Time Stamp Upload Key Module
- Time Stamp File Key Module
- Third Party Auditor(TPA) Module
- Cloud Module

MODULES DESCRIPTION

Client Module

- This module includes the Client registration and client login details.
- Every Client need to register while accessing to the cloud.
- Every Client will be activated by the Cloud.
- After Cloud activated, every Client need to provide time stamp upload key to upload a new files into cloud.
- Time stamp upload key will be provided by third party auditor.
- Client need to download the time stamp upload key when client uploading new files into cloud.
- Client can view file details and download the file using time stamp file key provided by TPA.

Time stamp upload key:

- Time stamp upload key will be provided by TPA. Client can download the upload key each time client uploading new file into cloud and they need not to give request key from TPA.
- At the time of client downloading the time stamp upload key, the request will send in directly to TPA and update according to time by TPA and send encrypted upload secret key to client.
- And finally, client can decrypt download the upload secret key.
- After getting decrypt upload secret key, now Client can upload a new file into cloud.

Time stamp file key:

- Each time client accessing and downloading the file from cloud, TPA will provide each time file update key to client registered mail Id. So same file key will not be there for same file.
- It will send as file time stamp update key, so corresponding client can use this file from different server without any other use of hacker or attacker.
- If Client again login with same server or different server, same file key will not been used by Client to download the file for more security.

Third Party Auditor (TPA) Module

- It acts as admin.
- TPA Provide time Upload secret key in Encrypted state for every client to upload new file into cloud. It will be send as in directly while Client downloading the upload key.
- The upload secret key, while user downloading key it will updated according to time.
- After cloud given auditing proof then only TPA can audit all files.
- And also provide the File Stamp key for all files to the client request for corresponding files key.

Cloud Module

- Activate data client.
- Cloud sends storage auditing proof for all files to TPA.
- Cloud can view the client downloaded files from cloud.

VI.CONCLUSION

In this paper, we study on how to outsource key updates for cloud storage auditing with key-exposure resilience. We propose the first cloud storage auditing protocol with verifiable outsourcing of key updates. In this protocol, key updates are outsourced to the TPA and are transparent for the client. In addition, the TPA only sees the encrypted version of the client's secret key, while the client can further verify the validity of the encrypted secret keys when downloading them from the TPA. We give the formal security proof and the performance simulation of the proposed scheme.

VII.REFERENCES:

[1] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. E. Spafford, "Secure outsourcing of scientific computations," *Adv. Comput.*, vol. 54, pp. 215–272, 2002.

[2] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in *Proc. 6th Annu. Conf. Privacy, Secur. Trust*, 2008, pp. 240–245.

[3] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 820–828.

[4] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," in *Proc. 17th Eur. Symp. Res. Comput. Secur.*, 2012, pp. 541–556.

[5] G. Ateniese *et al.*, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 598–609.

[6] A. Juels and B. S. Kaliski, Jr., "PORs: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 584–597.

[7] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2008, pp. 90–107.

[8] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw.*, 2008, Art. ID 9.

[9] F. Sebe, J. Domingo-Ferrer, A. Martinez-balleste, Y. Deswarte, and J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1034–1038, Aug. 2008.

[10] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiplereplica provable data possession," in *Proc. 28th IEEE Int. Conf. Distrib. Comput. Syst.*, Jun. 2008, pp. 411–420.

[11] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Efficient provable data possession for hybrid clouds," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 756–758.

[12] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," *IEEE Netw.*, vol. 24, no. 4, pp. 19–24, Jul./Aug. 2010.

[13] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 847–859, May 2011.

[14] K. Yang and X. Jia, "Data storage auditing service in cloud computing: Challenges, methods and opportunities," *World Wide Web*, vol. 15, no. 4, pp. 409–428, 2012.

[15] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu, "Dynamic audit services for outsourced storages in clouds," *IEEE Trans. Services Comput.*, vol. 6, no. 2, pp. 227–238, Apr./Jun. 2013.

[16] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1717–1726, Sep. 2013.

[17] H. Wang, "Proxy provable data possession in public clouds," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 551–559, Oct./Dec. 2013.

[18] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.

[19] B. Wang, B. Li, and H. Li Oruta, "Oruta: Privacy-preserving public auditing for shared data in the cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 43–56, Jan./Mar. 2014.

[20] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, 2009, pp. 213–222.

[21] B. Wang, B. Li, and H. Li, "Public auditing for shared data with efficient user revocation in the cloud," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2904–2912.

[22] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1717–1726, Aug. 2015.

[23] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1167–1179, Jun. 2015.