Grenze International Journal of Engineering and Technology, June Issue

A Novel Approach with Deep Learning Method with Effective Storage Security in Hybrid Clouds

Vijay Prakash[1], Aditya Tripathi[2], Shashank Saxena[3] and Arshad Ali[4]
[1-4]iNurture-TMU
vijayprakash15@gmail.com, aditya250790@gmail.com, shashankmtechcse@gmail.com, arshad.ali@inurture.co.in

***Abstract*— Platforms, data storage, and IT services are delivered over the Internet in cloud computing, a contemporary computing technology. Task management is crucial for effective scheduling and affects the overall effectiveness of cloud computing environments due to the full availability of resources and the significant number of tasks assigned to it. In cloud environments, security is a crucial concern in addition to timing. Since cloud computing services go beyond data archiving and backup, supporting data dynamics through the most popular types of data manipulation, like block modification, insertion, and deletion, is also crucial for practical use. A step, that is. Public auditability or dynamic data manipulation have not always been effective in prior attempts to ensure remote data integrity, but this document accomplishes both. We first recognized the challenges and potential security concerns of direct extension with fully dynamic data updates from prior work, and then we seamlessly incorporated these two crucial features into the protocol design. In particular, we enhance existing proof-of-storage models by modifying the conventional Merkle hash tree structure for block tag authentication to achieve effective data dynamics. This demonstrates how to construct an elegant validation scheme for dot. To secure cloud data storage, a variety of techniques have been put into practice [1]. The safety analysis method described in [1] is not a useful technique, though. The new idea of smart card authentication is used in this work to provide security for cloud data storage. Data storage in the cloud can be made more secure using an effective method called smart card authentication. We implemented this prototype in accordance with the CPDP scheme within the virtualization framework of a cloud-based storage service. Hadoop Distributed File System (HDFS) 6 is illustrated in Figure 5 as an example.**
It is a distributed, scalable, and portable file system [14]. HDFS's architecture is made up of NameNodes and DataNodes, where NameNodes translate filenames to a collection of block indices and DataNodes hold actual data blocks. The NameNode's index hash table and metadata must be integrated in order to support the CPDP scheme and provide query services based on hash values ((3)i,k) or index hash records (i). implement a protocol for verification.

***Index Terms*— cloud computing, cloud security, hybrid clouds, public verifiability, and storage security.**

I. INTRODUCTION

Hybrid clouds effectively offer dynamic scaling of services and data transfers by integrating a variety of private and public cloud services. For instance, a customer can combine data from various private or public providers into one backup file or archive (see Figure 1). As an alternative, a service can take data from other services that are located in a private cloud and store it in its own storage, creating a hybrid cloud.

Grenze ID: 01.GIJET.9.2.332

A Data-Proven (PDP) scheme based on public cloud offers a publicly accessible remote interface for exploring and managing vast amounts of data, but the benefit of the suggested PDP scheme is that performance is comparable to that of hybrid cloud. is capable of meeting the specific bandwidth requirements of, but the capacity cannot keep up with the time. Consider the hybrid cloud storage service depicted in Figure 1 as a solution to this problem. It consists of three distinct entities. an organization that manages and offers storage services through the use of large amounts of computing power and storage space. trusted third parties (TTPs) that keep track of and offer data retrieval services for customer review information.

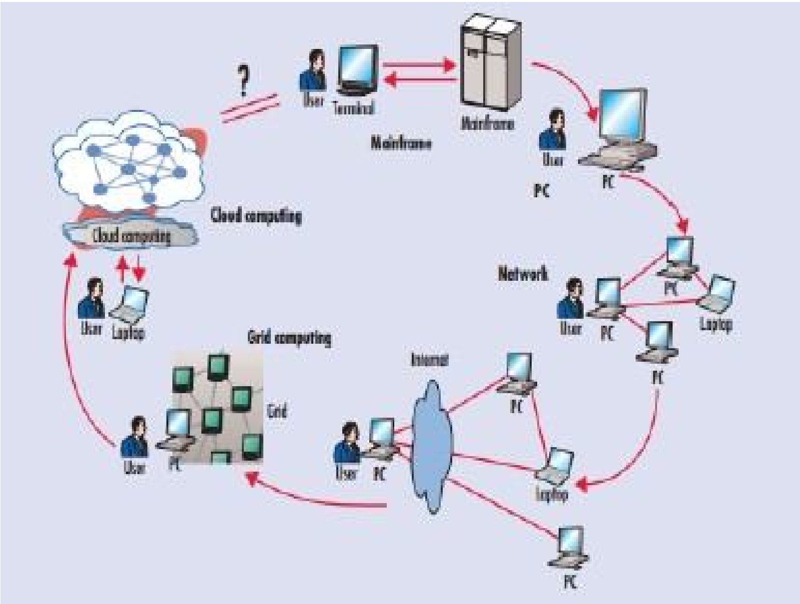
The new paradigm of data storage in the "cloud" is seen as a promising service platform for the Internet, but it raises a number of challenging design issues that have a big impact on the overall security and performance of an organization. The most significant issue with cloud data storage is data authentication on unreliable servers.

For example, a storage service provider may choose to conceal data failures from its clients in order to benefit from them. More importantly, by not keeping or consciously deleting data files that they routinely share with their customers, service providers can save money and disk space. The customer may find it practical to perform routine consistency checks without using a local copy of the data files to mitigate the issue when a significant amount of electronic data is offloaded and the customer has constrained resource capacity. must find a solution. There may be deep roots.

Several solutions have been put forth among various protection models and schemes to address this issue. A lot of time and effort is put into designing solutions in all of this work to satisfy various requirements. High system efficiency, stateless acknowledgments, limitless query use, and irretrievable data are a few of these.

All of the methods so far presented can be divided into two groups based on the role of the verifier in the model: private verifiability and public verifiability. Although a system with private verifiability can achieve higher system efficiency, public verifiability does not imply that only the customer (the data owner) can compete with cloud server.

Customers can do this without using their own computing resources to contract out service evaluations to independent external auditors (TPAs). In the cloud, the client itself cannot be relied upon to maintain integrity. The addition of public verifiability to validation protocols, which are anticipated to play a more significant role in achieving economies of scale in cloud computing, seems more balanced in real-world applications. be divided. The conversation seems to be similar to the performance conversation. When performing validation, validators shouldn't need external data.



II. RELATED WORK

In Yan Zhu, Huaixi, and others (2010). [1] Demand ownership of verifiable data to guarantee data integrity. The hybrid cloud's data ownership scheme is one that is helpful and auditable. It offers service scalability, data migration, and collectively gathers and stores customer data. Because operating costs are lower, communication complexity can be reduced to minimum.

According to Qian Wang et al. Through dynamic data manipulation, [2] introduces a new scheme that enables remote he data integrity and auditability. This step first pinpoints specific scaling issues and potential security concerns with fully dynamic data updates. Through manipulation of the conventional Merkle Hash Tree (MHT) structure, which is used to validate blocktags, we achieve efficient data dynamics and enhance retrievability models. This is a very effective and safe technique [2].

An allocation framework driven by models was put forth in 2012 by Tekin Bicer, David Chiu, and Gagan Agrawal [3]. Data-intensive applications running in hybrid cloud environments can benefit from this technique's support for time- and money-efficient execution. With a 3 point 6 percent error rate, you can meet implementation deadlines, stick to budgetary restrictions, and shorten application run times.

Using the currently in use cloud computing organisms, Haoming Liang, Wenbo Chen, and Kefu Shi [4] proposed a method for analyzing programming and task scheduling models. The programming process, its modification process, and the flow of replacing services and resources are all explained with the help of examples.

In 2010, Ravi Sandhu, Raj Boppana, and Ram Krishnan put forth a fresh idea for integrating mission-driven presentation, pliability, and security policies into the computing and communication infrastructure by integrating hooks and supporting protocols into the cloud. This methodology can effectively address the twin cloud security and accessibility issues [5].

A dynamic user-integrated cloud computing architecture was introduced in 2011 by Guannan HU and Wenhao ZHU [5]. This model expands the capabilities of cloud computing data centers by actively integrating clients with storage and computing capability. Services are offered to other users through client cooperation with the data center [5]. In order to better meet the practical learning needs of lifelong learners, Xiang Li, Jing Liu, Jun Han, and Qian Zhang proposed The article describes design of micro-learning platform architecture constructed through cloud computing expertise, details the layered structural design of micro-learning platform based cloud, and details the intention [6].

Xinwen Zhang, Anugeetha Kunjithapatham, Simon Gibbs, Joshua Schiffman, and Sangoh Jeong proposed A Solution for Authentication and Secure Session Management between Weblets Running on the Device Side and Weblets in the Cloud in 2009 [7]. allows cloud weblets to access sensitive user data through external web services and offers protected migration. In business environments, it enables application integration between private and public clouds [8].

Year: 2010, Yan Zhu, Huaixi, et al. [9] suggests a hybrid cloud data ownership plan that facilitates data migration and service scalability. This opens up possibilities in which several cloud service providers collaborate to store and manage customer data. Less overhead and simpler communication are the outcomes of this plan.

According to Qian Wang et al. [10] is a protocol that outlines the challenges and potential security concerns of direct extension with fully dynamic data updates before demonstrating how to create complex validation schemes for error-free integration. is recommending. This has an effect on block tag validation using the conventional Merkle Hash Tree (MHT) structure [10].

Arash Nourian and Muthucumaru Maheswaran proposed a new image coding scheme in 2012 that uses the Ima coding method, which transforms images using chaos maps, to enhance image privacy and enable the cloud to carry out some types of computation. introduced. chosen following random masking.

Jia Yu, Rajkumar Buyya, and Kotagiri Ramamohanarao presented a method in 2008 for allocating the proper resources to workflow tasks in order to complete their execution and enable each user to perform the desired function. It makes an effort to enhance the workflow scheduling algorithms currently in use that have been created and used by various grid projects [13].

Luis Mendonça and Henrique Santos published research findings and test results in 2012 that defined an efficient set of traffic parameters that could model both normal and abnormal behavior of networks and demonstrate abnormal and coordinated behavior. We focused on the detection of botnet movements in the instance of The detection framework model was also foreseen and tested with actual traffic gathered at the University of Minho campus edge [15].

A new security load balancing architecture based on multilateral security (LBMS) was proposed by Pengfei Sun Qingni Shen, Ying Chen Zhonghai, and Wu Cong Zhang in 2011. This architecture provides the ideal physical security device by automatically migrating tenant VMs during peak loads. CloudSim, a simulation of cloud computing, is the foundation of this method. When VMs move to physical machines for load balancing, this design tries to prevent potential attacks.

A new hybrid scheme that combines anomaly and signature detection with honeypots was put forth by Pragya Jain and Anjali Sardana in 2012. To enable real-time system capabilities, the first stage used signature-based detection of known worm attacks. Anomaly detectors can quickly spot deviations from the norm at the second level. Honeypots are used to identify zero-day attacks at the top level. It provides the advantages of a resource-efficient honey farm by utilizing honeypots and both detectors. Regulators route data traffic to the proper honeypots [18].

III. Problem Statement

The issue is that a variety of authentications have been used to protect dynamic data while numerous techniques have been used to store it. Although public verifiability and data dynamics are already implemented in cloud environments, it is still inefficient to provide both at the same time for remote data authentication. We also don't combine these technologies. Merkle hash tree storage of data failed. In order to provide a joint PDP scheme that supports dynamic scaling across multiple storage servers, a technique was put into place.

IV. Proposed Methodology

Smart cards can support a wide range of applications used by many different communities, including electronic (digital) signatures, email and data encryption, virtual private network (VPN) authentication, and password management. Biometric authentication and secure wireless network access. There are several variations of smart card technology. connect Hadoop and unstructured data.

Plastic cards (contact or contactless communication), USB devices, or as protected components that can be embedded in mobile phones and other devices are all examples of these. The following diagram displays the proposed method's architectural overview.

The suggested approach here uses smart cards to provide security while implementing the hybrid cloud concept.

A. Authentication using a smart card.

Configuration:.

Generation of keys in Module 1.
Choose a number of data.
Saves information as a Merkle hash tree.
Make a count of the files.
To facilitate authentication, make a private key.
Key Assignments for Module 2.
Files should be given keys.
Put the right key in the file's encryption.
In a hash table, keep the key and the information.
since using an index to access data is a simple process.
You can only search the data index; you cannot search the entire database. So, it will move along quickly.
Module 3: Cloud Server Data Storage.
These encrypted files should be saved to a different location on your cloud server.
Requestor possesses only appropriate keys.
The requester sends these keys to an outside verifier.
TPA uses these keys to validate data later on.
But the original data is hidden from TPA. Only uses cryptographic signature schemes for verification checks.
Integrity Check is a module in 4.
Decrypt all of the cloud server's files.
Put all files together.
Verify the data size. The size is identical to the initial data.
Put the corresponding encrypted file in this location in case data loss occurs as a result of a file's technical difficulties.
Keeps the entire file encrypted so that your security is never compromised.
Data Dynamics, Module 5.
At runtime, this module executes a few operations on the cloud server.
Change in data.
Incorporating data.
Erasure of data.
Batch Audit is the sixth module.

On cloud servers, many users keep their files.
In a batch system, each user validates his own data.
To accomplish this, a number of scheduling and priority algorithms are employed. E.g. bottleneck, deadlock.
therefore, the auditing time will be very less.

To determine if the evidence passes an integrity check, a verifier (TTPA) runs an algorithm. If the validation process worked, she returns TRUE; otherwise, she returns FALSE. Finally, the user receives a thorough test report. In the system description, we discussed the use of TTPA for hybrid cloud data integrity verification.
Her three main client-side phases make up our plan. The initialization stage, the key generation stage, and the tag generation stage are what they are called.

Step 1:

A certificate is first obtained by the user (or data owner) during his KeyGen stage.
CA (Certificate Authority) request and.
give his name and public key. tick The.
The format of certificates issued by CA is as follows:
C(DO) = [ID(DO),, sigCA(ID(DO),,)] -.
where [sigCA(ID (DO),,)] denotes the CA's digital signature. produced using a proprietary algorithm.

Step 2:

The data owner uses the verification algorithm verCA[C (DO)] to confirm permissions.

Step 3:

TTPA and the cloud server must receive this certificate in a secure manner. A perpetrator might pose as a trustworthy person.
user and acquire her valid.
public key and identity information are contained in a certificate. In order to prevent this, the consumer gives cloud servers and verifiers access to the certification authority verCA[C(DO)]'s verification algorithm. Online server.
additionally got a certificate.
C(DO) = [ID(DO), verDO, sigCA (ID(DO), verDO,)] ---(9).
C(CS) = [ID(CS), verCS, sigCA (ID(DO)), verCS,]----- (10).

Step 4: During the exchange of communications.

the and the TTPA.
Using the cloud server, a safe, authenticated channel is created.
the session key. These two parties communicate using the same session key.
The generated proof (P), which Cloud Server signed with its private key.
creating a signature with (SK).
Next, send (P,, C.
(DO) to TTPA.

Step 5:

After receiving the signature, TTPA first verifies the certificate using a verification algorithm. If.
is certified, the answer is checked using its public key.
(PK). Otherwise, the consistency check will stop.

Step 6:

Decrypt the proof reply once more in the following step.
utilizing the public key (PK). After making sure that, this is done.
Verification of the certificate of authenticity. Her.
The authenticity of the server is then established by verifying the signature.

Step 7:

Finally, TTPA checks the proof by running the.
verification algorithm. By doing this, you can prevent active attackers from altering your data in any way.

V. RESULT ANALYSIS

TABLE I. PROPOSED WORK FROM VARIOUS ATTACKS

Recurring Attacks	Y
Attacks involving identity theft	Y
Insider dangers	Y
External assaults	Y
Listening in	Y
Identity Theft	Y
Attacks using passwords	Y

The various proposed attack preventions are being criticized, as shown in Table 1.

TABLE II. SUGGESTIONS FOR AUTHENTICATION FACTORS.

Number of bits in token	Number of bits in secrete value	Time taken
64	216	24.5 sec

An analysis of first factor authentication is presented in Table 2. Here, the number of bits used to generate the secret value is dependent on the number of bits used to create the token.

TABLE III. THE SUGGESTED METHOD'S MEMORY COMPARISON.

Storage/ scheme	Our scheme	R. song al et.
Smart card	640 bits	480 bits
Server	320 bits	640 bits

It shows that the proposed method reduces the load on the server because the server only stores the server's private key.

VI. CONCLUSION

In any environment, security is crucial to the transmission of data from sender to receiver. When data is consistent, it is kept in a single place across many nodes. For data storage and data integrity, the cloud concept was introduced. The ability to check nodes from one cloud to another and back up dynamic data is typically possible if multiple clouds are implemented and data is stored dynamically. Add unstructured data to challenging Hadoop. By authenticating the node from one cloud to another, we use the hybrid cloud concept to dynamically store data exposed from any node in any cloud. Describe the idea. We're working to make it a reality. Data ownership is supposed to be dynamic, so whenever one of your node's clients wants to access data kept in another cloud, it has to first approve public access to the data. The suggested method used here improves authentication, lessens the chance of eavesdropping, and guards against a number of attacks like DOS, replay attacks, etc.

VII. REFERENCES

[1] Yan Zhu, Huaixi Wang, Zexing Hu, Gail-Joon Ahn, Hongxin Hu, Stephen S. Yau "Efficient Provable Data Ownership for Hybrid Clouds", 2010, 1st Part on Computer and Communication Security. Proceedings of the 17th ACM Conference (CCS) '10), S. 756-758, 2010.

[2] Qian Wang, Cong Wang, Jin Li1, Kui Ren and Wenjing Lou, "For Storage Security in Cloud Computing Enabling public verifiability and data dynamics of ", 2009 Proceedings of die 14. Europäische Konferenz zur Forschung in der Computersicherheit (ESORICS'09), S. 355-370, 2009.

[3] Tekin Bicer, David Chiu und Gagan Agrawal, „Time and Cost Sensitive Data-Intensive Computing on Hybrid Clouds“, 2012 IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 636 – 643, Mai 2012.
[4] Haoming Liang, Wenbo Chen and Kefu Shi “Cloud Computing: Programming Models and
[5] Guannan HU und Wenhao ZHU, "A Dynamic User-integrated Cloud Computing Architecture", 2011 International Conference on Innovative Computing and Cloud Computing (ICCC '11), S.: 36-40, 2011.
[6] Xiang Li, Jing Liu, Jun Han and Qian Zhang, "Architecture Design of Microlearning Platform Based on Cloud Computing", Proceedings of the 2011 International Conference on Innovative Computing and Cloud Computing (ICCC '11), S. 80- 83, 2011
[7] Zhang, Joshua Schiffman, Simon Gibbs, Anugeetha Kunjitha, Patham and Sangoh Jeong, Securing Elastic Applications on Mobile Devices for Cloud Computing, Proceedings of the 2009 ACM Workshop on Cloud Computing Security (CCSW '09), S. 127-134, 2009. Yan Zhu, Huaixi Wang, Zexing Hu, Gail-Joon Ahn, Hongxin Hu, Stephen S. Yau, “Efficient Provable Data Ownership for Hybrid Clouds,” Proceedings of the 17th ACM Conference on Computer and Communication Security (CCS '10), S. 756-758, October 2010.
[8] Qian Wang, Cong Wang, Jin Li1, Kui Ren and Wenjing Lou, “Enabling Public Verifiability and Data Dynamics of Storage Security in Cloud Computing,” Proceedings of the 14th European Conference on Research in Computer Security (ESORICS'09), S. 355-370, 2009.
[9] Arash Nourian und Muthucumaru Maheswaran, "Towards privacy-enhanced limited image processing in the cloud", 13th ACM/IFIP/USENIX International Middleware Conference Proceedings of the 9th Middleware Doctoral Symposium (MIDDLEWARE '12), Artikel Nr. 5, 2012.
[10] Jean Bacon1, David Evans1, David M. Eyers1, Matteo Migliavacca2, Peter Pietzuch2 and Brian Shand3, “End-to-Work in the Cloud”. Enhancing End Application Security," Proceedings of the ACM/IFIP/USENIX 11th International Conference Middleware (Middleware '10), S. 293-312, 2010.
[11] Jia Yu, Rajkumar Buyya, Kotagiri Ramamohanarao, "Workflow Scheduling Algorithm for Grid Computing", 2008 Springer Berlin Heidelberg, ISSN NO. 1860-949X, pp. 173–214,
[12] Jon Oberheide, Kaushik Veeraraghavan, Evan Cooke, Jason Flinn and Farnam Jahanian, "Virtualized Security Services in the Cloud for Mobile Devices", Proceedings of the First Workshop on Virtualization in Mobile Computing (MobiVirt' 08), S. 31-35, 2008.
[13] Luís Mendonça und Henrique Santos, ``Botnets: A Heuristic-Based Detection Framework", Proceedings of the Fifth International Conference on Security of Information and Networks (SIN '12), S. 33-40, 2012
[14] Alex Kantchelian Justin Ma und Ling Huang "Robust detection of comment spam using entropy rate", Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence (AISec '12) Record, S. 59-70, 2012.
[15] Pengfei Sun Qingni Shen, Ying Chen Zhonghai und Wu Cong Zhang, "POSTER: LBMS: Load Balancing based on Multilateral Security in Cloud", Proceedings of the 18th ACM Conference on Computer and Communication Security. (CCS '11), S. 861-864, 2011.
[16] Shiuan-Tzuo Shen, Hsiao-Ying Lin and Wen-Guey Tzeng, "Effective Alignment for Secure Erasure Code-Based Storage Systems Reliability Check Scheme", IEEE Trans. On Reliability, vol. 64. Nr. September 3, 2015.