Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

PUBLIC AUDITING OF SHARED DATA IN CLOUD WHILE MAINTAINING CONFIDENTIALITY

K. HARSHINI PG Scholar, Dept of CSE, Quba College of Engineering & Technology, Venkatachalam, AP, India.

M.HYMAVATHI Assistant Professor, Dept of CSE, Quba College of Engineering & Technology, Venkatachalam, AP, India.

ABSTRACT

Data sharing among numerous users as well as cloud storage are both prevalent with cloud data services. Sadly, there are hardware/software failures, human errors, and other factors that raise doubts about the integrity of cloud data. Several procedures have been created to make it possible for both data owners and public verifiers to effectively verify the accuracy of cloud data without having to download all of the data from the cloud server.Yet, using these current systems for public auditing on the integrity of shared data will ultimately lead to the disclosure of private information—identity—to public verifiers. We suggest a novel privacy-preserving approach in this research that enables public auditing of shared data kept in the cloud.We use ring signatures in particular to compute the verification information required to audit the accuracy of shared data. With the help of our approach, public verifiers can effectively check the integrity of shared data without having to download the full file, while the identity of the signer on each block in private data is kept secret from them..Additionally, rather than validating each auditing activity individually, our mechanism may carry out several auditing tasks at once. Our test findings show that our technique is effective and efficient at verifying the integrity of shared data.

1. INTRODUCTION

The field of computer science is drawn to the appeal of cloud computing. The cloud will revolutionise the IT industry, according to the Gartner report. The cloud is transforming our way of life by offering users new kinds of services. Cloud users receive services without being concerned with the specifics. Cloud computing, according to NIST, is a model that enables ubiquitous, practical, on-demand network access to a shared pool of reconfigurable computing resources (such as networks, servers, storage, applications, and services) that can be quickly provisioned and released with little management work or service provider interaction.People are becoming more and more interested in cloud computing. Although load balancing is receiving a lot of attention from researchers, cloud computing is effective and scalable, but maintaining the stability of processing so many operations in the cloud computing environment is a highly complicated problem.

Workload control is essential for load balancing issues since the job arrival pattern is unpredictable and each cloud node has a different capacity, which affects system performance and stability. Depending on whether system dynamics are crucial, load balancing strategies can either be

static or dynamic. While dynamic schemes will add to the system's costs but can alter when the system status changes, static schemes don't use system information and are less complex. This uses a dynamic system because of its adaptability. The model has balancers and a core controller to gather and evaluate the data. Thus, the dynamic control has little influence on the other working nodes. The system status then provides a basis for choosing the right load balancing strategy.

The public cloud, which consists of several nodes with distributed computing resources spread across numerous geographical regions, is the target of the load balancing paradigm presented in this article. As a result, this architecture creates several cloud divisions from the public cloud. These divisions make load balancing easier in vast and complicated environments. The cloud's main controller selects the appropriate partitions for incoming jobs, and the balancer for each partition selects the ideal load-balancing method.

1.1 MOTIVATION

The performance of the entire cloud will be enhanced by effective load balancing. Unfortunately, there isn't a universal approach that can be applied to any conceivable variety of circumstances. In order to enhance current solutions and address new issues, numerous techniques have been developed. Each technique is advantageous in a specific context, but not always. As a result, the current model combines many approaches and alternates between them depending on the system status. A more complex method can be employed for the normal state, whereas a very simple method can be utilised for the partition idle state.As the status changes, the load balancers then adjust their techniques. Here, the normal state makes use of a load balancing technique based on game theory while the idle status makes use of an upgraded Round Robin algorithm..

1.2 PROBLEM DEFINITION

Several computing resources are accessible and only a small number of workloads are arriving while the cloud division is not in use. In this case, a straightforward load balancing strategy can be applied because this cloud division has the capacity to process workloads as soon as feasible. There are numerous straightforward ways for balancing the load, including the Random algorithm, the Weight Round Robin, and the Dynamic Round Robin. Because to its simplicity, the Round Robin algorithm is utilised here. One of the simplest load balancing algorithms is the Round Robin algorithm, which routes each new request to the server after it in the queue. The algorithm lacks status information because it doesn't keep track of each connection's status.Every node in the standard Round Robin algorithm has an equal chance of being chosen. The configuration and performance of each node will differ in a public cloud, therefore this strategy may overwhelm some nodes. As a result, "Round Robin based on the load degree evaluation"—an upgraded version of Round Robin—is adopted. The algorithm is still quite straightforward. The load balancing table's nodes are arranged from lowest to highest according to load degree prior to the RoundRobin stage. The system creates a circular queue and repeatedly moves

through it. Then, tasks will be allocated to nodes with low load indices. Whenever the balancer updates the Load Status Table, the nodeorder is modified.

Therefore, during the refresh time T, there can be inconsistent read and write operations. When the balance table is updated, a job arriving at the cloud partition at this time will result in an inconsistent situation. The information will still be outdated even though the system status has changed..

This could result in the selection of an incorrect load strategy and the incorrect order of nodes. Two load status tables—Load Status Tables 1 and 2—should be constructed in order to remedy this issue. Each table is also given a flag to designate Read or Write. This table is used by the Round Robin algorithm based on the load degree evaluation when the flag is set to "Read.".

When the flag is set to "Write," this table is refreshed and new data is added. As a result, while the other table is being created with the most recent information, one table continuously provides the right node placements in the queue for the enhanced Round Robin algorithm. The flag for the table is changed to "Read" when the data has been updated, while the flag for the other table is changed to "Write". To resolve the contradiction, the two tables then alternate..

4.3 A load-balancing plan for the default state A different approach to load balancing is employed since when the cloud partition is functioning normally, workloads are arriving much faster than when it is idle and the situation is far more complicated. Each user wants his tasks to be finished as quickly as possible, hence the public cloud needs a method that can finish all users' tasks with a respectable response time..

For distributed systems, Penmatsa and Chronopoulos presented a static load balancing technique based on game theory. Also, this paper offers us a fresh analysis of the load balance issue in the cloud context. The load balancing process used in the cloud computing environment can be seen as a game because it is a distributed system implementation. Cooperative and non-cooperative games can be found in game theory. When playing cooperative games, the decision-makers eventually reach a binding agreement.Each decision-maker weighs their options before making a choice. Each player in a non-cooperative game takes choices solely for his or her personal advantage. As soon as the system enters the Nash equilibrium, each decision-maker takes the best choice. Each player in the game chooses a strategy, and the Nash equilibrium occurs when no player can gain from altering that strategy while the other players' strategies remain the same..

1.3 OBJECTIVE OF PROJECT

Each decision-maker weighs their options before making a choice. Each player in a non-cooperative game takes choices solely for his or her personal advantage. As soon as the system enters the Nash equilibrium, each decision-maker takes the best choice. Each player in the game chooses a strategy, and the Nash equilibrium occurs when no player can gain from altering that strategy while the other players'

strategies remain the same..

Yet, the integrity of data saved in the cloud is under question because it is simple for data to be lost or corrupted owing to technology malfunctions and human mistake [1]. It is best to carry out public auditing by bringing in a third party auditor (TPA), who offers its auditing service with more potent computing and communication capacities than regular users, in order to protect the integrity of cloud data..

The first PDP mechanism [2] to carry out public auditing is intended to verify the accuracy of data stored on an unreliable server without obtaining the complete data set. Moving on, Wang et al. [3] (referred to as WWRL in this study) is designed to create a public auditing method for cloud data, so that during public auditing, the third party auditor is not made aware of the content of private data belonging to a personal user..

We think that one of the most compelling characteristics that encourages cloud storage is the ability to share data among numerous users. How to protect identify privacy from the TPA has arisen as a unique issue as a result of the public auditing procedure for shared data in the cloud.

2. LITERATURE SURVEY

1. On predictive models and user drawn graphical passwords

AUTHORS: P. C. van Oorschot and J. Thorpe

Users frequently select passwords that are simple to remember, have patterns, and are therefore susceptible to brute-force dictionary attacks when using text-based password schemes. This prompts us to wonder if users' propensity to select memorable passwords makes other password types (such graphical ones) susceptible to dictionary attacks as well. For systems where passwords are generated exclusively from a user's memory, we offer a technique to anticipate and model a number of such classes. These classes, according to our theory, define weak password subspaces usable by attack dictionaries. We employ this technique in conjunction with cognitive studies on visual memory for user-drawn graphical passwords. We are motivated by these cognitive studies to develop a set of password complexity variables (e.g., reflective symmetry and stroke count), that specify a group of classes. We use the "Draw-A-Secret" (DAS) graphical password scheme of Jermyn et al. [1999] as an example to better comprehend the magnitude of these classes and, consequently, how vulnerable the password subspaces they define may be. We examine the size of these classes for DAS under practical parameter choices and demonstrate that they can be coupled to generate surprisingly few, unexpectedly popular subspaces, with bit sizes ranging from 31 to 41. (58 bits). Our findings statistically back up claims that user-drawn graphical password systems apply safeguards such proactive password validation and graphical password rules or recommendations..

2. Modeling user choice in the PassPoints graphical password scheme
AUTHORS: A. E. Dirik, N. Memon, and J.-C. Birget

We design a model to determine where users are most likely to click while making graphical passwords in the PassPoints system. A user selects a number of points from an image that is shown on the screen as their PassPoints password. We can anticipate the entropy of a click point in a graphical password for a given image using our model, which forecasts probabilities of possible click points. The model enables us to analyse potential word attacks against the PassPoints system and automatically determine whether a given image is suitable for the system. We contrast the outcomes of trials involving human participants with the predictions made by our model.At this stage, our model and the experiments are small and limited; but they show that user choice can be modeled and that expansions of the model and the experiments are a promising direction of research.

3. Securing passwords against dictionary attacks
AUTHORS: B. Pinkas and T. Sander

The usage of passwords presents a significant point of vulnerability in computer security since dictionary assaults using automated algorithms can frequently guess passwords with ease. Despite their well-known security flaws, passwords continue to be the most popular authentication technique. User authentication is undoubtedly a real-world issue. This issue needs to be resolved from the standpoint of a service provider while taking into account practical limitations like the available hardware and software infrastructures. From the viewpoint of the user, user-friendliness is a crucial criterion. In this study, we propose a novel password authentication method that keeps the benefits of traditional password authentication while tenfold increasing the costs of online dictionary assaults..The suggested method for enhancing the security of user authentication schemes is simple to implement and does away with some of the issues raised by earlier suggested solutions. Our main concept is to effectively combine conventional password authentication with a challenge that is (nearly) impossible for automated systems to crack using dictionary attacks but is very simple to solve for human users. This is accomplished without impairing the system's usability. Moreover, the suggested strategy offers improved defence against denial-of-service assaults on user accounts..

3. PROBLME STATEMENT

Public auditing is a term used to describe a number of technologies that enable both the data owner and a third party to easily undertake integrity checks without downloading all of the data from the cloud. These algorithms retrieve a random combination of the blocks rather than the entire data during integrity checking. Data is separated into numerous small blocks, each of which is independently signed by the owner. A third-party auditor (TPA) who can do expert integrity checks on data or a data user (such as a researcher) who wants to use the owner's data via the cloud could both serve as public verifiers..In order to prevent any public verifiers from seeing the content of a personal user's private data during public

auditing on cloud data, Wang et al. created an advanced auditing technique. Sadly, the aforementioned public auditing tools are solely concerned with cloud-based personal data. One of the most interesting elements that encourages cloud storage, in our opinion, is the ability to share data between numerous users. Thus, it is equally essential to guarantee the accuracy of shared data integrity in the cloud. In fact, shared data integrity can be confirmed by extending the capabilities of current public auditing tools. Yet, a new substantial privacy concern that has arisen when using existing ways to communicate data is the leakage of identity privacy to public verifiers.

3.1 LIMITATION OF SYSTEM

If identity privacy on shared data is not maintained during public auditing, critical confidential information will be made available to public verifiers. To protect identify privacy during public auditing, it is crucial and important to protect these personal information..

4. PROPOSED SYSTEM

In this research, we present Oruta, an unique privacy-preserving public auditing system, to address the aforementioned privacy problem on shared data. More specifically, we utilise ring signatures to construct homomorphic authenticators in Oruta, so that a public verifier is able to verify the integrity of shared data without retrieving the entire data while the identity of the signer on each block in shared data is kept private from the public verifier. Furthermore, we expand our method to accommodate batch auditing, which can carry out several auditing activities at once and enhance the effectiveness of verification for numerous auditing tasks..Random masking, which has been used in WWRL and can protect data privacy from outside verifiers, is compatible with Oruta. To enable dynamic data, we also use index hash tables from a previous open auditing solution. Oruta and existing mechanisms are compared on a high level..

ADVANTAGES OF PROPOSED SYSTEM:

A public verifier can accurately assess the integrity of shared data. Throughout the auditing process, a public verifier cannot tell who signed each block in the shared data. The generated ring signatures can support blockless verifiability as well as the preservation of identity privacy..

5. IMPLEMENTATION

This work is specifically focused on a public cloud but there are other cloud computing categories. A service provider offers services over a public cloud, which is based on the typical cloud computing model. A big public cloud will have plenty of nodes, and the nodes will be spread out over the world. To

manage this huge cloud, cloud partitioning is required. A cloud partition is a section of the public cloud that is divided into regions according to locales..

The system chooses which cloud partition should be given the task, under the direction of the main controller. After that, the partition load balancer chooses how to distribute the jobs among the nodes. This partitioning can be carried out locally while a cloud partition's load status is normal. This task needs to be moved to another partition if the cloud partition load status is abnormal..

The primary controller and the balancers work together to find a load-balancing solution. Prior to communicating with the balancers within each cloud partition to update this status information, the main controller first distributes jobs to the appropriate cloud partition. Smaller data sets will result in faster processing speeds because the primary controller deals with information specific to each partition. Each partition's balancers collect status information from each node and then select the best method for distributing the jobs..

When a job arrives at the public cloud, the first step isto choose the right partition. The cloud partition status can be divided into three types:

(1) Idle: When the percentage of idle nodes exceeds α, change to idle status.
(2) Normal: When the percentage of the normal nodesexceeds _, change to normal load status.
(3) Overload: When the percentage of the overloaded nodes exceeds , change to overloaded status.

The node load degree is related to variousstatic parameters and dynamic parameters. The static parameters include the number of CPU's, the CPU processing speeds, the memory size, etc. Dynamic parameters are the memory utilization ratio, the CPU utilization ratio, the network bandwidth, etc. The load degree is computed from these parameters as below:

Step 1: Define a load parameter set: F= {F1,F2,…Fm} with each Fi(1<=i<=m, ∈ [**0**, **1**]) Fi parameter being either static or dynamic. M represents the total number of the parameters.

Step 2 : Compute the load degree as:

Load degree(N)=$\sum_{i=1}^{m} \alpha i \; Fi$,

αi($\sum_{i=1}^{m} \alpha i$ =1) are weights that may differ fordifferent kinds of jobs.N represents the current node.

Step 3 : Define evaluation benchmarks. Calculate the average cloud partition degree from the node load degree statistics as:

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

Load degreeavg $= \frac{\sum_{i=1}^{n} \text{Load degree(Ni)}}{n}$

The bench mark Load degreehigh is then set fordifferent situations based on the Load degreeavg.

Step 4 Three nodes load status levels are then defined as:

Idle When Load_degree(N)= 0, there is no job being processed by this node so the status is charged to Idle**.**

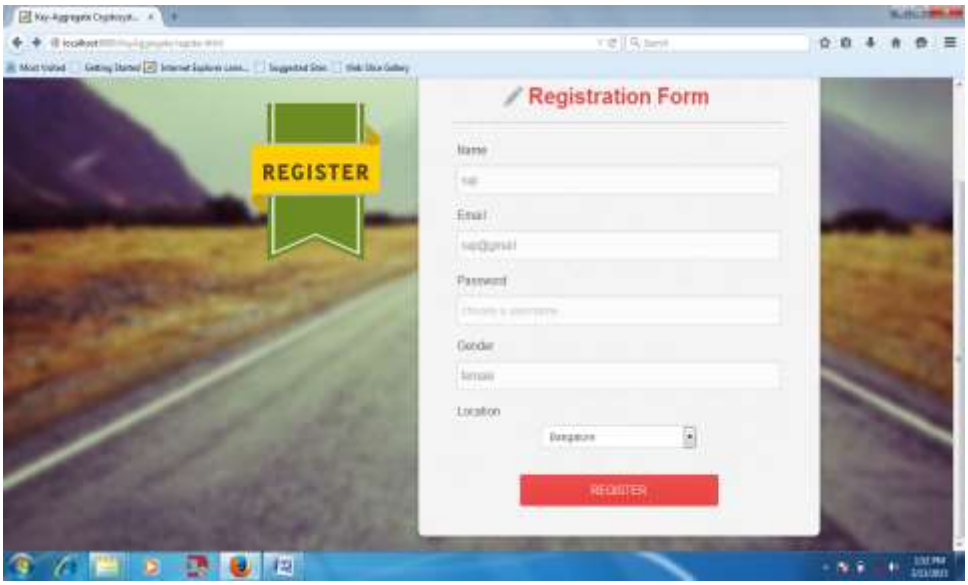
NormalFor0<Load_degree(N)<=Load_degreehigh, the node is normal and it can process other jobs.

Overloaded When

> Load_degreehigh<=Load_degree(N),the node is not available and can not receive jobs untilit returns to the normal.

The cloud partition balancers construct Load StatusTables and input the load degree results into them. A Load Status Table is maintained by each balancer and is updated once per fixed period T. The balancers utilise the table to determine the partition status after that. There is a separate load balancing approach for each partition status. The balancer sends a job to the nodes based on its current load strategy when it arrives at a cloud partition. The balancers alter their approach in response to changes in the cloud partition status.

6. OUTPUT SCREENS



Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

7. CONCLUSION & FUTURE ENHANCEMENT

Since this work just provides a conceptual framework, additional effort will be required to put the framework into use and address unresolved issues. Some key points include:

(1) Rules for cloud division The division of clouds is a challenging issue. The framework will therefore require a thorough cloud division technique. Nodes in a cluster, for instance, could be far from one another or there might be clusters in the same region that are yet far apart. Simple geographic location should serve as the basis for the division rule (province or state).

(2) How to configure the refresh period: The main controller and the cloud partition balancers must update the data at regular intervals for the data statistics analysis. The system performance will be impacted by the high frequency if the period is too short. The knowledge will be too dated if the time frame is too extended to make wise decisions. In order to establish an acceptable refresh period, tests and statistical methods are required..

In this article, we suggest Oruta, the first public auditing tool for shared data in the cloud that protects privacy. With Oruta, the TPA can easily audit the consistency of shared data without being able to tell who signed each block, protecting user identify privacy. How to effectively audit the integrity of shared data with dynamic groups while yet protecting the identity of the signer on each block from the third party auditor is an intriguing problem that will be addressed in our future work..

REFERENCES

[1]M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Kon-winski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, April 2010.

[2]G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores, " in *Proc.ACM CCS*, 2007, pp. 598–610.

[3]C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in *Proc. IEEEINFOCOM*, 2010, pp. 525–533.

[4]R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret, " in *Proc. ASIACRYPT*. Springer-Verlag, 2001, pp. 552–565.

[5]D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate an d Verifi-ably Encrypted Signatures from Bilinear Maps," in *Proc. EUROCRYPT*. Springer-Verlag, 2003, pp. 416–432.

[6]H. Shacham and B. Waters, "Compact Proofs of Retrievabili ty," in *Proc.ASIACRYPT*. Springer-Verlag, 2008, pp. 90–107.

[7]Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Publi c Verifiability and Data Dynamic for Storage Security in Cloud C omput-ing," in *Proc. European Symposium on Research in Computer Security*. Springer-Verlag, 2009, pp. 355–370.