International Journal of Engineering & Technology, 7 (2.32) (2018) 477-480

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Framework for Data Integrity Through Lineage Tracking in Cloud

Divya Vadlamudi[1], Dr. K. Thirupathi Rao[2], Sravani Bodempudi[3], Lavanya Kadulla[4]

[1,2]Professor, Department of Computer science and Engineering, KL Education Foundation, Guntur.
[3,4]Department of Computer Science and Engineering,KL Education Foundation, Guntur.
**Corresponding author E-mail: 1divya.movva@kluniversity.in*

Abstract

Cloud Computing [1] is a service oriented paradigm that pursuits at sharing resources to a large number of tenants. It's far approximately using internet to save, control and process facts in preference using personalized computer. Even though Cloud computing has the pre-requisite to store the data; some users find some issues regarding safe data Storage in cloud. In that case metadata is the data that helps to determine the history of the particular data object we opted starting from its original resources. Here not only integrity of the data is proved but the specific lineage of data is also proved. Lineage is useful to know the origin of the data and its transformations in the cloud. In this paper we develop a framework to ensure Access control and data integrity using lineage of data stored in the cloud[2], and some general semantic definitions for integrity properties of lineage tracking. The main contribution of this paper is to explore framework through metadata for integrity and access control.

Keywords: *Cloud Computing, Integrity, Meta data, Access control*

1. Introduction

Cloud computing lets in customers and endeavours with numerous computing centres to get the techniques and (retrieve, get admission to, alter) records both in personal cloud, or on third-party server that is located within the cloud and hence making the customer to access the mechanisms and data for extra efficiency and safety purpose. Cloud storage[3] is equal as records garage where in the digital statistics is stored in logical puddles and the cloud storage companies are chargeable for preserving the records to be handled and ingress for the people. Customers and firms tend to purchase or rent the garage from the contributors to keep customers firms or application statistics. We give credence with the aid of stepping in the cloud structure with all provenance [5] requirements and present the performance effects of the structure.

2. Literature Review

We have seen about integrity and metadata. Metadata [4] may be created manually and automatically. However manual introduction tends to be extra accurate. Whereas computerized metadata is much greater essential, only showing records that includes the record size, report extension by the user who created the file.
Metadata is nothing but provenance. Although both security and provenance have their own huge areas of concern, there is a very significant intersection. Classic conceptions of data security refer to integrity, access control and confidentiality. Access control is important security aspect and this can be defined as restricting access for people by keeping login passwords and credentials for protecting data from unauthorized people. Authentication is done by verifying user's or host's identity to get access for their data. Authorization is giving or restricting access based on roles of user.

Data integrity is the assurance, that information is unrestricted and can only be accessed or modified by those authorized users. Integrity involves maintaining the consistency, accuracy and trustworthiness of data over its entire lifecycle.

Towards Network Provenance

1. Starting point as distributed provenance [5]: Routing algorithm is the best example we can give for this scenario. Now here we have a group of enhancements to provide integrity for the queries that are on the unregistered statements. We apply this concept mainly in HTTP requests that are connected to the DNS server due to network mis-configurations.

a. The provenance graph [7] is been divided and is distributed to many nodes. For instance a tuple is stored at another node and that node is to be indicated as location symbol. The other side of the version is user has no control of statistics. There are issues of statistics and integrity. The provider level policies and compliances are completely controlled by the carrier issuer.
b. This provenance graph is of dynamic-packets. After that scenario the routing entries will be inserted or updated.
c. For example, we consider SPAN deployments for scale industries with less network connections at any node and that nodes provide us a large amount of data.

2. Time awareness: The only key concern in this append is maintenance and queries. In order to address this we maintain our provenance[20] at an active state depending on the desired trade-off. The main disadvantage of this model is that it is very expensive model that chooses automatically the best approach for the design we ensured. And the external resources are shared among

multiple users, and the IT security issues are keen and data is vulnerable to thefts.

3. Negative Provenance: This case is related mainly as "why not" queries that we store in our database that explain a particular tuple that is been absent in the query result. The negative we see in this model is that they share the similar roles with proxy tuples or co-tuples.

3. Proposed Solution

Provenance in cloud computing usually gives us the information that "needs to show the history of that object", starting from its actual resources. The information in this could be sensitive and also give a framework [6]. It also shows us the standards of the results and fortifies the number of iterations it has performed till the expe riment is done successfully.

SaaS: Use of SaaS programs has a tendency to overcome the ability of the software ownership by using casting for the need of specialized committee to manage deploy and upgrade software. In addition to reduce the cost of software program, SaaS programs are typically supplied in a subsidy version.
PaaS: As we have the maximum extent of cloud offerings, PaaS is constructed on the top level of generation. Groups can handle resources as they use them in the place of creating another investment in hardware with redundant resources.
Examples of PaaS carriers include heroku, google search engine.
IaaS: IaaS clients provide the cloud servers and also their related assets via dashboard API. They have direct access to their servers and cloud garage. Users of IaaS can outsource and construct a "digital information centre" within the cloud and feature to get right of entry for building technologies and useful resource skills of the conventional information.

4. Provenance in Data Integrity

The trouble of proving information integrity in cloud computing is with the resource of proving a scheme through which customers are in a position to check the integrity [8] of their data stored within the cloud. For this motive we employ a relatively new concept in the cloud called "data provenance". Our scheme is successful to lessen the "third party services" and provide additional hardware guide to the replication.

In order to check the integrity for our data, we need to initially store the data in and further requirements are shown below for integrity process and showing the final results.

Step 1:

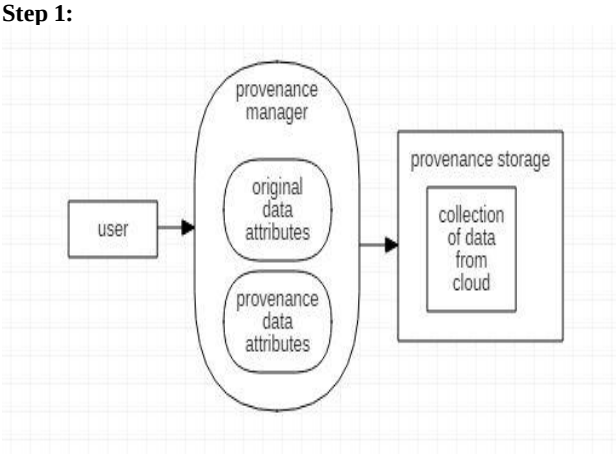

Fig 1.1: User storing his data in cloud.

In the first step when the user gives the data it goes to provenance manager [10].Then it is divided into provenance data and original data, and provided attributes are ensured to that data. Then it enters into provenance storage where it collects all the data from cloud.

Prov Manager:

a. **Possibility:** The data can be transferred to any program. Additionally the procedure is designed to sketch the data into another cloud database.
b. **Light Weight:** It stores only the provenance data that is noteworthy for the process to be designed.
c. **Productive look:** Procedure algorithms are implemented to give required forage mechanisms.

Step2:

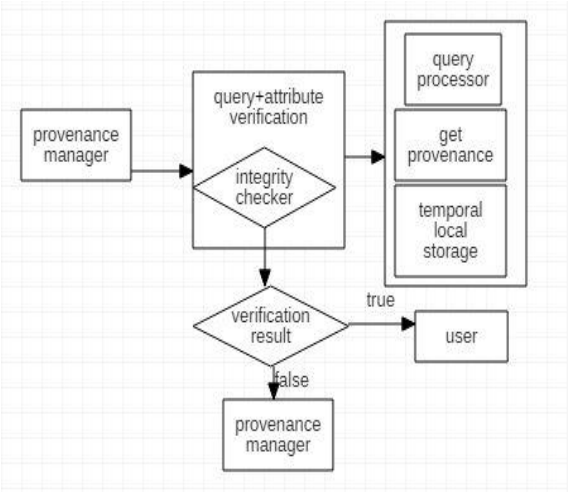

Fig 1.2: Integrity tracking process

After getting our data into provenance manager there comes the integrity checker which consists of both query and attribute verification, i.e., the queries we give for processing stage and the attributes that are mentioned for provenance data and original data in step 1.Here in this verification process it involves query processor, get provenance, temporal local storage and we finally get the result of that process.

Integrity tracker: This section is done on the idea of provenance statistics for checking if any violations are present. It executes via developing and publishing an internet provider as integrity carrier to the prevailing cloud environment. The general procedure avoids if any TPA or any hardware aid for integrity checking.

Query processor: Query processor[9] accepts requests from cloud clients for the verification of records integrity. The request query is generated based at the user individual choice. The choice can be truly a document call or combination of several parameters which includes the admission to control the queries. Therefore, various parameters are set through the consumer side to process the request query. Such alternatives are made to be held to cease client so that user can customize the request query and consequent statistics for the integrity is decided on content material.

Get provenance: The get provenance [11] aspect accepts the query from query processor. The get provenance has an interface to the provenance. This issue executes the query and extracts the related statistics primarily based on the customers' choice which includes the information that includes product, identity, quantity, name of the proprietor, closing accesses time, and the ultimate acknowledged length of the records. Those extracted consequences are then saved in the garage where in the subsequent issue can use it for computing the integrity evidence.

Step 3:

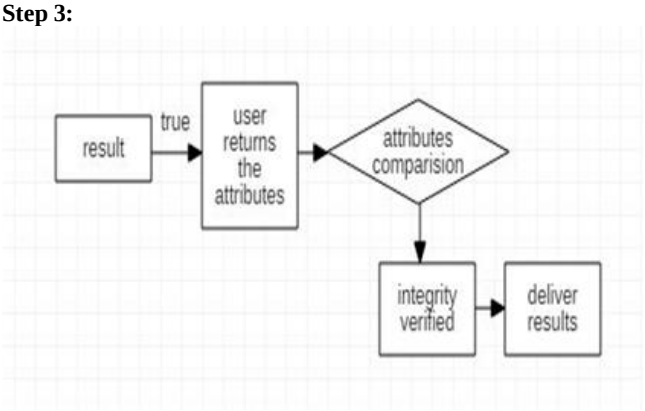

Fig 1.3: Delivering the results to end user.

After getting the integrity result if it is true then the user returns the attributes. Then it checks if the attributes with the user and the attributes with cloud matches then we say that our integrity is been verified and it gives us the final results.

Compute Proof: The compute evidence is simple in every key components of the information that is been provided by the user. While the provenance information is been retrieved from a consumer query, compute evidence is executed for producing a integrity evidence. This is obtained via evaluating the provenance records extracted in get provenance level with the metadata[8] of the authentic items stored in cloud. The very last consequences of the compute evidence suggest whether the integrity of the precise information product is violated or not. In case of integrity violation it additionally tracks the process and shows us the result by whom the document has been modified or changed. The effects are forwarded to the supply consequences detail.

Step 4:

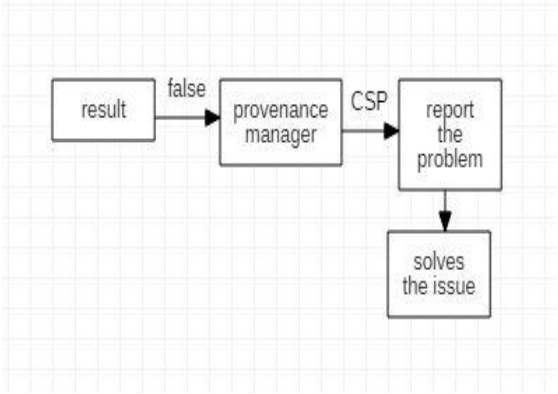

Fig 1.4: Reporting problem to provenance manager with the help of CSP.

If the result is false then it goes to the provenance manager and reports the problem with the help of cloud service provider (CSP) and there the problem will be solved and the process of integrity checking shown in fig 1.2 will be processed again until we get the final result.

Deliver Results:

Deliver outcomes is the remaining factor of the integrity provider. It may liable for converting the outcomes of generated evidence into a costumer readable shape. It communicates with all the clients and executes through graphical customer interface. Then it goes with the drift of records in the information integrity.

5. Data Lineage Tracking in Cloud for Integrity

The concept of data lineage[12] in intended to make the process that we design the easier one. It involves the elements in such a way that each row of data in the output is unambiguously to the source we give as the input. It includes all the transformation process during all its travel until the execution has been completed.

Usage of lineage tracking:

1. Trustworthiness
2. Easier trouble shooting
3. Expose leaky process
4. Visibility

1. **Trustworthiness:** When the origin of each row of data and the path it took to arrive is systematically tracked, users and administrators of the data will have more reason to trust the data.

2. **Easier trouble shooting**: When the ETL data path is self-describing[19]. It makes testing and trouble shooting for easier. It analyses and solves the serious problems.

3. **Expose leaky process**: When each row of data is tractable from source to destination, it helps to reveal any holes (errors) in the process where the data might lose that is stored.

4. **Visibility:** As we cannot count as many times the computer is asking. So, that we take the help of the client to find and give a final documentation for the deal in the ETL process.

Properties:

- This cloud sharing and facility what it offers is coupled with the sheer wide variety of users make cloud environments liable to critical protection dangers.
- Lineage requires a semantic framework that makes a few affordable assumptions approximately the subjective components that are concerned within the design.
- Clients require their facts to be secure and personal from an unauthorized get admission.
- Numerous algorithms and protocols are applied by numerous components of this model to provide most level of integrity.
- Provenance[14] is metadata that describes the facts of an object provenance is an critical factor within the verification, audit trails, reproducibility, privacy and safety, accept as true with, and reliability in lots of fields ranging from the artwork.

Advantages

Controls: Higher controls for statistics, customers and data belongings.

Safety: As cloud belongs to a single user, therefore, the design and the components are been configured to provide maximum extent of safety.

Performance: Typically non-public clouds[13] are placed in the internal part of the design we developed and the web services ensures the efficiency[15] and provides high network performance.

Clean customization: The hardware [16] and different assets can be customized effortlessly via the agency.

Simple and clean: Public clouds are available as a carrier in the file systems, they are clean to deploy.

Cost: Initial investment is very low.

Less time: Resources and offerings are in the right away for saving time.

6. Conclusion

In this paper, we mentioned the statistics of integrity, especially in cloud environments [17]. Ensuring the process of guarantee in on premise cloud platform that is similarly in cloud garage. Our proposed scheme is primarily based totally on using provenance and that may be a nearby resource to the cloud environment. This metadata is carried out in our method to provide integrity that leaks throughout the implementation of product life cycle in cloud. In our proposed solution we made a clear view that only if the attributes that are with the user entity and the attributes with the cloud entity matches in the process of integrity tracking then itself the user can proceed with his further steps or else the user need to give a report to the provenance manager to get issue solved. Hence we ensure that this methodology provides wide range of security and the performance is up to the level. In this research, we investigated what methods have to be implemented on the recorded information provenance[18] to reap our predicted effects.

7. Future Scope

Here in this paper we have given the framework[6] that is to be followed for the data that is stored in our cloud database and need to be secured at high performance level. And in the next level we need to implement a code in Linux showing the above framework with all perfect provenance queries for integrity through lineage tracking as of ETL process.

References

[1] "Mell, P. and Grance, T., 2011. The NIST definition of cloud computing. "

[2] Hashemi, S.M. and Bardsiri, A.K., 2012. Cloud computing vs. grid computing. ARPN journal of systems and software, 2(5), pp.188-194.

[3] Luo, W. and Bai, G., 2011, September. Ensuring the data integrity in cloud data storage. In Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on (pp. 240-243). IEEE.

[4] Simmhan, Y.L., Plale, B. and Gannon, D., 2005. A survey of data provenance in e-science. ACM Sigmod Record, 34(3), pp.31-36.

[5] Simmhan, Y.L., Plale, B. and Gannon, D., 2005. A survey of data provenance techniques. Computer Science Department, Indiana University, Bloomington IN, 47405.

[6] Simmhan, Y., Plale, B., Gannon, D. and Marru, S., 2006. Performance evaluation of the karma provenance framework for scientific workflows. Provenance and Annotation of Data, pp.222-236.

[7] Abbadi, I.M., 2013. A framework for establishing trust in Cloud provenance. International journal of information security, 12(2), pp.111-128..

[8] Bates, A., Mood, B., Valafar, M. and Butler, K., 2013, February. Towards secure provenance-based access control in cloud environments. In Proceedings of the third ACM conference on Data and application security and privacy (pp. 277-284). ACM.

[9] de Oliveira, D., Ocaña, K.A., Baião, F. and Mattoso, M., 2012. A provenance-based adaptive scheduling heuristic for parallel scientific workflows in clouds. Journal of Grid Computing, pp.1-32.

[10] Asghar, M., Ion, M., Russello, G. and Crispo, B., 2012. Securing data provenance in the cloud. Open problems in network security, pp.145-160..

[11] Glavic, B. and Dittrich, K.R., 2007, March. Data Provenance: A Categorization of Existing Approaches. In BTW (Vol. 7, No. 12, pp. 227-241).

[12] P. Buneman, s. khanna, and w. chiew tan, "why and in which: a characterization of records provenance," in icdt '01: lawsuits of the 8th worldwide conference on database principle. springer, 2001, pp. 316–330.

[13] de Oliveira, D., Baiao, F.A. and Mattoso, M., 2010. Towards a taxonomy for cloud computing from an e-science perspective. In Cloud Computing (pp. 47-62). Springer London.

[14] Davidson, S.B. and Freire, J., 2008, June. Provenance and scientific workflows: challenges and opportunities. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp. 1345-1350). ACM.

[15] Cheah, Y.W. and Plale, B., 2012 IEEE 8th International Conference on E-Science (e-Science).

[16] Moreau, L., Freire, J., Futrelle, J., McGrath, R.E., Myers, J. and Paulson, P., 2008, June. The open provenance model: An overview. In International Provenance and Annotation Workshop (pp. 323-326). Springer, Berlin, Heidelberg.

[17] Jung, I.Y. and Yeom, H.Y., 2011. Provenance security guarantee from origin up to now in the e-science environment. Journal of Systems Architecture, 57(4), pp.425-440.

[18] Suen, C.H., Ko, R.K., Tan, Y.S., Jagadpramana, P. and Lee, B.S., 2013, July. S2logger: End-to-end data tracking mechanism for cloud data provenance. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on (pp. 594-602). IEEE.

[19] Zhou, W., Sherr, M., Tao, T., Li, X., Loo, B.T. and Mao, Y., 2010, June. Efficient querying and maintenance of network provenance at internet-scale. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data(pp. 615-626). ACM.

[20] Cheney, J., 2011, June. A formal framework for provenance security. In Computer Security Foundations Symposium (CSF), 2011 IEEE 24th (pp. 281-293). IEEE.