Enabling Identity-based Integrity Verification Data Sharing Model using TPA and Batch Auditing Protocol

T Poongothai[1],
[1]Professor,
Department of Computer Science and Engineering
K.S.R. College of Engineering
Tiruchengode, India

S. Prakashraj[2], E. Preetha[3], J. Priyangaa[4]
[2,3,4] UG Students
Department of Computer Science and Engineering
K.S.R College of Engineering
Tiruchengode, India

***Abstract*: In cloud storage services, users store their data remotely to the cloud and realize the data sharing with others. In Electronic Health Records (EHRs) system, the cloud file might contain some sensitive information. The sensitive information should not be known to others when the cloud file is shared. Encrypting the whole shared file realizes the sensitive information hiding, but will make this shared file unable to be used by others. How to realize data sharing with sensitive information hiding in remote data integrity auditing still has not been explored up to now. In order to address this problem, this paper proposes a remote data integrity auditing scheme that realizes data sharing with sensitive information hiding. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. This paper proposes a secure cloud storage system supporting privacy-preserving public auditing and extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. In addition, it articulates performance optimization mechanisms for this scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. It shows that the solution introduces lower computation and communication overheads in comparison with non-cooperative approaches**.

Keywords— Auditing protocol, Cloud storage, Data sharing, TPA.

I. INTRODUCTION

Cloud computing is an attracting technology in the field of computer science. It is proven that cloud will bring changes to the IT industry. The cloud is changing our life by providing users with new types of services. Users get service from a cloud without paying attention to the details. 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. More and more people pay attention to cloud computing. Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers. In this paper, the following four architectural patterns are distinguished:**

a. Replication of applications allows to receive multiple results from one operation performed in distinct clouds and to compare them within the own premise. This enables the user to get an evidence on the integrity of the result.

b. Partition of application System into tiers allows separating the logic from the data. This gives additional protection against data leakage due to flaws in the application logic.

c. Partition of application logic into fragments allows distributing the application logic to distinct clouds. This has two benefits. First, no cloud provider learns the complete application logic. Second, no cloud provider learns the overall calculated result of the application. Thus, this leads to data and application confidentiality.

d. Partition of application data into fragments allows distributing fine-grained fragments of the data to distinct clouds. None of the involved cloud providers gains access to all the data, which safeguards the data's confidentiality.

Each of the introduced architectural patterns provides individual security merits, which map to different application scenarios and their security needs. Obviously, the patterns can be combined resulting in combined security merits, but also in higher deployment and runtime effort. The following sections present the four patterns in more detail and investigate their merits and flaws with respect to the stated security requirements under the assumption of one or more compromised cloud systems.

The main objective of this paper is,

- To set different trust level is set to different cloud providers and encryption/decryption is varied based on the clouds computational capability.
- To take partial data of files from multiple mirror locations and send to selected client.
- To reduce the risk for data and applications in a public cloud is the simultaneous usage of multiple clouds.

- To handle Irrelevant size blocks of data among the multiple cloud service providers based on their computational capabilities.

II. RELATED WORKS

Thomas Ristenpart and Eran Tromer [1] the authors stated that third-party cloud computing represents the promise of outsourcing as applied to computation. Services, such as Microsoft's Azure and Amazon's EC2, allow users to instantiate virtual machines (VMs) on demand and thus purchase precisely the capacity they require when they require it. In turn, the use of virtualization allows third-party cloud providers to maximize the utilization of their sunk capital costs by multiplexing many customer VMs across a shared physical infrastructure. However, in this paper, the authors showed that this approach can also introduce new vulnerabilities. This paper explores the practicality of mounting such cross-VM attacks in existing third-party compute clouds. The attacks they considered require two main steps: placement and extraction [2]. Placement refers to the adversary arranging to place their malicious VM on the same physical machine as that of a target customer.

Juraj Somorovsky and Mario Heiderich [3] the authors refer to two distinct classes of attacks on the two main authentication mechanisms used in Amazon EC2 and Eucalyptus cloud control interfaces. The first class of attacks complies of the XML Signature Wrapping attacks (or in short signature wrapping attacks) on the public SOAP interface of the Cloud. They demonstrated that these control interfaces are highly vulnerable to several new and classical variants of signature wrapping. For these attacks, knowledge of a single signed SOAP message is sufficient to attain a complete compromization of the security within the customer's account. The reason for this easiness is that one can generate arbitrary SOAP messages accepted by this interface from only one valid signature. To make things even worse, in one attack variant, knowledge of the (public) X.509 certificate alone enabled a successful execution of an arbitrary cloud control operation on behalf of the certificate owner.

Sven Bugiel and Stefan Nürnberger [4] In this paper they considered security and privacy aspects of real-life cloud deployments, independently from malicious cloud providers or customers. They focused on the popular Amazon Elastic Compute Cloud (EC2) and give a detailed and systematic analysis of various crucial vulnerabilities in publicly available and widely used Amazon Machine Images (AMIs) and show how to eliminate them. Their Amazon Image Attacks (AmazonIA) deploy an auto-mated tool that uses only publicly available interfaces and makes no assumptions on the underlying cloud infrastructure. They were able to extract highly sensitive information (including passwords, keys, and credentials) from a variety of publicly available AMIs.

George Danezis and Benjamin Livshits [5] the authors stated that privacy is considered one of the key challenges when moving services to the Cloud. Solution like access control is brittle, while fully homomorphic encryption that is hailed as the silver bullet for this problem is far from practical. But would fully homomorphic encryption really be such an ective solution to the privacy problem? And can we already deploy architectures with similar security process. They proposed one such architecture that provides privacy, integrity and leverages the Cloud for availability while only using cryptographic building blocks available today.

Stephan Grob and Alexander Schill [6], the authors stated that cloud computing, i. e. providing on-demand access to virtualised computing resources over the Internet, is one of the current mega-trends in IT. Today, there are already several providers offering cloud computing infrastructure (IaaS), platform (PaaS) and software (SaaS) services. Although the cloud computing paradigm promises both economical as well as technological advantages, many potential users still have reservations about using cloud services as this would mean to trust a cloud provider to correctly handle their data according to previously negotiated rules.

Martin Burkhart and Mario Strasser [7], describe a secure multiparty computation (MPC) allows joint privacy-preserving computations on data of multiple parties. Although MPC has been studied substantially, building solutions that are practical in terms of computation and communication cost is still a major challenge. In this paper, they investigated the practical usefulness of MPC for multi-domain network security and monitoring. They first optimized MPC comparison operations for processing high volume data in near real-time. They then designed privacy-preserving protocols for event correlation and aggregation of network traffic statistics, such as addition of volume metrics, computation of feature entropy, and distinct item count.

Sven Bugiel and Stefan Nurnberger [8], proposed an architecture and protocols that accumulate slow secure computations over time and provide the possibility to query them in parallel on demand by leveraging the benefits of cloud computing. In their approach, the user communicates with a resource-constrained Trusted Cloud (either a private cloud or built from multiple secure hardware modules) which encrypts algorithms and data to be stored and later on queried in the powerful but untrusted Commodity Cloud.

Ristenpart et al. [14] exhibited coarser, cross-VM, access-driven side-channel attacks on modern symmetric multi-processing (SMP, also called multi-core) architectures. But their attack could only provide crude information (such as aggregate cache usage of a guest VM) and, in particular, is insufficient for extracting cryptographic secrets. Despite the clear potential for attacks, no actual demonstrations of fine-grained cross-VMside-channels attacks have appeared. The oft-discussed challenges to doing so stem primarily from the facts that

VMMs place more layers of isolation between attacker and victim than in cross-process settings, and that modern SMP architectures do not appear to admit fine-grained side-channel attacks (even in non-virtualized settings) because the attacker and victim are often assigned to disparate cores.

III. METHODOLOGY

In this paper, the Proficient Privacy Protection Scheme (PPPS) is proposed to provide the appropriate privacy protection which is satisfying the user-demand privacy requirement and maintaining system performance simultaneously[9]. At first, the privacy level is analyzed by users those require and quantify security degree and performance of encryption algorithms. Then, an appropriate security composition is derived by the results of analysis and quantified data. Finally, the simulation results show that the PPPS not only fulfills the user-demand privacy but also maintains the cloud system performance in different cloud environments [10].

The proposed system covers multiple cloud service provider environments. In addition, size blocks of data are being processed with varying size nature in different cloud locations having same copy of data. The data blocks is stored and retrieved in different cloud locations based on the storage and computational capability. Thus the proposed system explores such issue to provide the support of variable-length block verification.Likewise, the privacy level for all cloud providers is analyzed by trusted authority and security degree and performance is quantified for encryption algorithms [11].

A. PRIVATE KEY GENERATION PROCESS AND VERIFICATION BY USERS

The private key generation (PKG) process chooses two multiplicative cyclic groups of prime numbers. The PKG randomly chooses an element x belongs to those prime numbers. The PKG computes the public value and the master secret key. The PKG publishes system parameters, i.e, these prime numbers list and holds the master secret key. During communication with the users, after receiving the user's identity ID, he PKG randomly picks a value from prime numbers list and computers the private key of the user ID [12]. The PKG sends it to the user ID. The user ID verifies the correctness of the received private key by using the system parameters given by PKG earlier. The user ID refuses the private key if not matches; otherwise it accepts the key.

B. ADD CLOUD NODE DATA

In this phase, the cloud node id and the cloud provider name is added. There are more cloud nodes for single cloud provider. From the trusted authority, the cloud node receives secret tags for file blocks so that the blocks can be processed/ verified by the cloud nodes [13].

C. PRIVACY PRESERVING AUDITING PROTOCOL

In this phase, the file name is selected, the file content is split into various segments and each segment is given two prime numbers each of which belongs to two prime order. One is given to user, other is given to third party auditor. The combination of the two is kept in server. During auditing, third party auditor randomly picks the segment ids and send corresponding prime number vector to cloud server. If the credentials match, then the file integrity is said to be verified [15].

D. BATCH AUDITING PROTOCOL

In this phase, during auditing, two processes of same third party auditor randomly pick the two set of segment ids and send corresponding prime number vectors to cloud server. If the credentials match, then the file integrity is said to be verified [16].

E. STORAGE AND COMPUTATIONAL CAPABILITY BASED FILE STORAGE

In this phase, the file content is selected from client files. The file data is saved in cache. Either DES (Data Encryption Standard) or AES (Advanced Encryption Standard) encryption work is carried out and the selected file is encrypted. The requirement of this level presents that no sensitive information in the data. Cloud location with low computational capability uses weak encryption composition (DES) and high computational capability uses more encryption (AES) to obtain more performance for using cloud services. Finally decryption work (DES and AES) is carried out [17].

IV. EXPERIMENTAL RESULTS

The Multi cloud model is a data replication algorithm based on the A-star best-first search algorithm with IAP model. The E-IDM -star starts from the null solution that is called a root node. The communication cost at each node n is computed as: [18]

$$cost(n) = g(n) + h(n)$$

where *g(n)* is the path cost for reaching *n* and *h(n)* is called the heuristic cost and is the estimate of cost from *n* to the goal node. The E-IDM-star searches all of the solutions of allocating a fragment to a node. The solution that minimizes the cost within the constraints is explored while others are discarded.[19]

S.NO	Cloud Node Communication	MCM COST		E-IDM Cost	
		G (n)	H (n)	G (n)	H (n)
1	25	8	0.8	5	0.5
2	50	14	0.14	12	0.12
3	75	24	0.24	20	0.20
4	100	32	0.32	27	0.27
5	125	45	0.45	39	0.39

Table 4.1 Secure Communication MCM and E-IDM Model

The table 4.1 show secure communication cloud node for MCM and E-IDM model. The table contains number of communication node, cost path (g(n)) values,

heuristic cost path (h(n)) for MCM and E-IDM model details are shows.

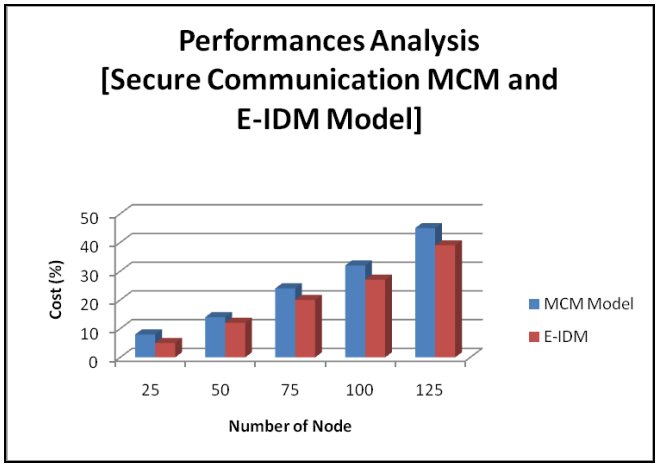

Fig 4.1 Secure Communication MCM and IE-IDM Model

The fig 4.1 and fig 4.2 show secure communication cloud node for MCM and E-IDM model. The figure contains number of communication node, cost path (g(n)) values, heuristic cost path (h(n)) for Multicloud Model and E-IDM model details are shows. The following resultst is evident the highest performance while the betweenness centrality showed the cloud performance

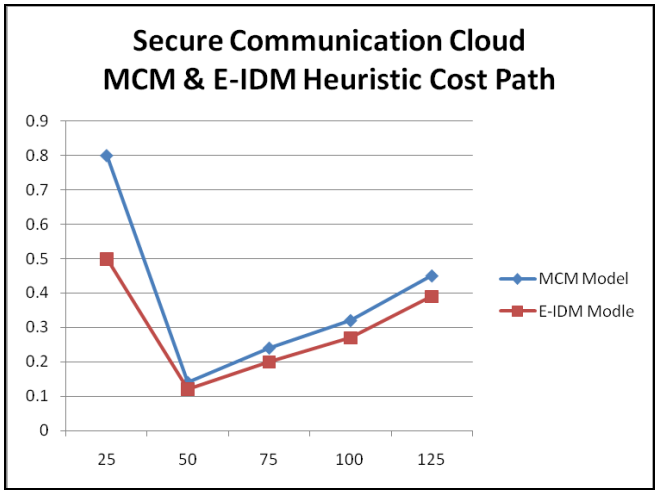

Fig 4.2 Secure Communication Cloud – MCM and E-IDM Heuristic Cost Path

- The proposed system provides a safe cloud storage methodology which supports privacy-preserving third party auditing better than existing system.
- This thesis suggests that the security can be increased if the architecture is changed from single cloud to multi cloud environment.
- Security mechanisms involved during third party auditing of outsourced data is discussed.
- The methods are studied to perform the auditing without demanding the local copy of data and thus drastically reduce the communication and computation overhead.
- Four schemes are presented that can be applied in multi cloud environment to increase the security aspects.
- Hiding resource usage statistics of a single resource for a single cloud provider is achieved if first method is applied.
- The computation and data transfer size is very low if the second method is applied.
- The third method provides the security such that a single provider may not be aware of the execution flow of the single application as well as the cloud provider could not know or access all the data.
- The fourth method provides the benefit of auditing with very low credential data to verify the file content.
- It is proved that the third party auditing computation time is better than existing approach.
- The future study should focus on security proof and enhancements in data retrieval of the proposed framework [20].

V. CONCLUSION

In this paper, the problem of secure communication is eliminated. In addition, the application required less working experience in systems to run the software. The application is tested well so that the end users use this software for their whole operations.

It is believed that almost all the system objectives that have been planned at the commencement of the software development have been net with and the implementation process of the project is completed. A trial run of the system has been made and is giving good results the procedures for processing is simple and regular order. The process of preparing plans been missed out which might be considered for further modification of the application. This work effectively stores and retrieves the records from the cloud space database server. The records are encrypted and decrypted whenever necessary so that they are secure.

The following enhancements are should be in future.

- ✓ The application if developed as web services, then many applications can make use of the records.
- ✓ The data integrity in cloud environment is not considered. The error situation can be recovered if there is any mismatch.
- ✓ The web site and database can be hosted in real cloud place during the implementation.

REFERENCES

[1] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds,” Proc. 16th ACM Conf. Computer and Comm. Security (CCS ’09), pp. 199-212, 2015.

[2] Y. Zhang, A. Juels, M.K.M. Reiter, and T. Ristenpart, “Cross-VM Side Channels and Their Use to Extract Private Keys,” Proc. ACM Conf. Computer and Comm. Security (CCS ’12), pp. 305-316, 2012.

[3] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, “All Your Clouds Are Belong to Us: Security Analysis of Cloud Management Interfaces,” Proc. Third ACM Workshop Cloud Computing Security Workshop (CCSW ’11), pp. 3-14, 2011.

[4] S. Bugiel, S. Nu ¨ rnberger, T. Po ¨ppelmann, A.-R. Sadeghi, and T.Schneider, "AmazonIA: When Elasticity Snaps Back," Proc. 18th ACM Conf. Computer and Comm. Security (CCS '11), pp. 389-400, 2011.
[5] G. Danezis and B. Livshits, "Towards Ensuring Client-Side Computational Integrity (Position Paper)," Proc. ACM Cloud Computing Security Workshop (CCSW '11), pp. 125-130, 2011.
[6] S. Groß and A. Schill, "Towards User Centric Data Governance and Control in the Cloud," Proc. IFIP WG 11.4 Int'l Conf. Open Problems in Network Security (iNetSeC), pp. 132-144, 2011.
[7] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos, "SEPIA: Privacy-Preserving Aggregation of Multi-Domain Network Events and Statistics," Proc. USENIX Security Symp., pp. 223-240, 2015.
[8] D. Hubbard and M. Sutton, "Top Threats to Cloud Computing V1.0," Cloud Security Alliance, http://www. cloudsecurityalliance.org/topthreats, 2010.
[9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. Commun. ACM, 53(4):50–58, 2010.
[10]R. Meushaw and D. Simard. A network on a desktop. NSA Tech Trend Notes, 9(4), 2010.
[11]P. England and J. Manferdelli. Virtual machines for enterprise desktop security. Information Security Technical Report, 11(4):193 – 202, 2016.
[12]T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh. Terra: a virtual machine-based platform for trusted computing. In ACM Symposium on Operating Systems Principles, pages 193–206. ACM, 2013.
[13]O. Acii¸cmez. Yet another microarchitectural attack: Exploiting I-cache. In ACM Workshop on Computer Security Architecture, pages 11–18, October 2017.
[14]T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In 16th ACM Conference on Computer and Communications Security, pages 199–212, 2016.
[15]Gnu Privacy Guard. www.gnupg.org, 2012.
[16]J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. Openpgp message format. Technical report, RFC 2440, November, 2000.
[17]McIntosh, M., and Austel, P. XML Signature Element Wrapping attacks and Countermeasures. In SWS '05: Proceedings of the 2005 workshop on Secure web services (New York, NY, USA, 2005), ACM Press, pp. 20-27.
[18] Nadalin, A., Kaler, C., Monzillo, R., and Hallam-Baker, P. Web Services Security: SOAP Message Security 1.1 (WS-Security 2004). OASIS Standard Specification (2006).
[19]McIntosh, M., and Austel, P. XML Signature Element Wrapping attacks and Countermeasures. In SWS '05: Proceedings of the 2005 workshop on Secure web services (New York, NY, USA, 2005), ACM Press, pp. 20-27.
[20]NIST. The NIST Definition of Cloud Computing (Draft). 2011. Special Publication 800-145 (Draft).

.