ABSTRACT

Client action logs can be an important wellspring of data in cloud criminological examinations; henceforth, guaranteeing the dependability and uprightness of such logs is urgent. Most existing answers for secure logging are intended for customary frameworks as opposed to the intricacy of a cloud situation. In this paper, we propose the Cloud Log Assuring Soundness and Secrecy (CLASS) process as an elective plan for the verifying of logs in a cloud situation. In CLASS, logs are encoded utilizing the individual client's open key with the goal that lone the client can unscramble the substance. So as to counteract unapproved alteration of the log, we produce evidence of past log (PPL) utilizing Rabin's unique finger impression and Bloom channel. Such a methodology diminishes confirmation time altogether. Discoveries from our investigations conveying CLASS in OpenStack exhibit the utility of CLASS in a genuine world context.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

ABBREVATION	EXPANSION
JSP	JAVA SERVER PAGE
HTML	HYPER TEXT MARKUP LANGUAGE
JDBC	JAVA DATABASE CONNECTIVITY
CSP	CLOUD SERVICE PROVIDER
PPL	PROOF OF PAST LOG

CHAPTER 1

INTRODUCTION

OVERVIEW

CLOUD COMPUTING

CLOUD capacity, security and protection are genuinely settled research regions [1-7], which isn't astounding thinking about the across the board appropriation of cloud administrations and the potential for criminal abuse (for example bargaining cloud records and servers for the taking of delicate information). Curiously however, cloud criminology [8-10] is a generally less gotten subject. If a cloud administration, cloud server, or customer gadget has been undermined or engaged with malignant digital movement (for example used to have unlawful substance, for example, radicalization materials, or lead appropriated disavowal of administration (DDoS) assaults) [11, 12], agents should almost certainly direct criminological investigation so as to "answer the six key inquiries of an occurrence–what, why, how, who, when, and where" [13]. Because of the inborn idea of cloud advancements, regular computerized measurable techniques and devices should be refreshed to hold a similar handiness and relevance in a cloud domain [14]. In contrast to a traditional customer gadget, cloud virtual machines (VMs) can be bolstered by equipment that may be found remotely and, in this manner, would not be physically available (for example out of the jurisdictional region) to an agent. What's more, VMs can be appropriated over various physical gadgets in a grouped situation or they can exist inside a pool of VMs on the equivalent physical parts. In this manner, holding onto the machine for scientific examination isn't reasonable in many examinations. Besides, information dwelling in a VM might be unstable and could be lost once the power is off or the VM ends. Henceforth, the cloud specialist co-op (CSP) assumes a critical job in the accumulation of evidential information (for example cloud client's action log from the log). For instance, the CSP composes the movement log (cloud log) for every client. Along these lines, counteracting alteration of the logs, keeping up a legitimate chain of guardianship and guaranteeing information security is critical [15]. This examination considers "action log information" as any recorded PC occasion that compares to a particular client. Such information must be kept up privately to preserver client security and to encourage potential insightful exercises.

In 2016, Zawoad et al. proposed a safe logging administration called "SecLaaS" [16] that is intended to gather information from at least one log sources, parse the information and after that store the parsed information in relentless capacity so as to moderate the hazard related with information unpredictability. Before the putting away of information, it encodes the log and creates a log chain to accomplish classification and respectability separately. SecLaaS scrambles the log(s) utilizing the examining office's open key and stores the encoded log(s) in a cloud server. This guarantees security and privacy of the cloud client, except if the specific client is liable to an examination (for example through a court request). To encourage log honesty, SecLaaS produces evidence of past log (PPL) with the log chain and distributes it openly after each predefined age. A trust model was additionally proposed that stores the PPL in different mists to limit the danger of a malignant cloud substance adjusting the log. In any case, in SecLaaS, it is hard to guarantee or confirm that the CSP is composing the right data to the log, or that any data relevant to the examination isn't precluded or altered. In particular, SecLaaS does not give the client the capacity to confirm the exactness of the log (since the log is encoded with the office's open key). At the end of the day, SecLaaS has impediments in tending to responsibility and straightforwardness authorized, particularly from the viewpoint of the client Broadening SecLaaS, we propose a safe cloud logging plan, Cloud Log Assuring Soundness and Secrecy (CLASS), intended to guarantee CSP responsibility (for example composing the right data to the log) and save the client's security–for example our commitment in this paper. In particular, we incorporate the capacity for the client to check the precision of their log. To do this, the log will be encoded utilizing the client's open key (as opposed to the organization's open key). To abstain from acquainting pointless deferrals with the legal examination, during client enrollment with the cloud administration, both the CSP and the client will all things considered pick an open/private key pair alluded to as substance covering key (CC-key) for the client. The comparing (content disguising) private key will be imparted to different CSPs utilizing Shamir's [17] or Blakley's [18] mystery sharing plans. This would enable the private key to be recovered at whatever point fundamental. We likewise exhibit how we can use Rabin's unique finger impression [19] and sprout channel in PPL age to set up log veracity. We at that point execute CLASS in OpenStack and assess its exhibition.

SOFTWARE DEVELOPMENT ENVIRONMENT

CLIENT SERVER

With the varied topic in existence in the fields of computers, Client Server is one, which has generated more heat than light, and also more hype than reality. This technology has acquired a certain critical mass attention with its dedication conferences and magazines. Major computer vendors such as IBM and DEC, have declared that Client Servers is their main future market. A survey of DBMS magazine revealed that 76% of its readers were actively looking at the client server solution. The growth in the client server development tools from $200 million in 1992 to more than $1.2 billion in 1996. Client server implementations are complex, but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server. The software mediating this client server interaction is often referred to as MIDDLEWARE. The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request. The key client server idea is that client as user is essentially insulated from the physical location and formats of the data needs for their application. With the proper middleware, a client input from or report can transparently access and manipulate both local database on the client machine and remote databases on one or more servers. An added bonus is the client server opens the door to multi-vendor database access indulging heterogeneous table joins.

WHAT IS CLIENT SERVER

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems. Both provide shared network access to data but the comparison dens there! The file server simply provides a remote disk drive that can be accessed by LAN applications on a file by file basis. The client server offers full relational database services such as SQL- Access, Record modifying, Insert, Delete with full relational integrity backup/ restore performance for high volume of transactions, etc. the client server middleware provides a flexible interface between client and server, who does what, when and to whom.

WHY CLIENT SERVER

Client server has evolved to solve a problem that has been around since the earliest days of computing: how best to distribute your computing, data generation and data storage resources in order to obtain efficient, cost effective departmental an enterprise wide data processing. During mainframe era choices were quite limited. A central machine housed both the CPU and DATA (cards, tapes, drums and later disks). Access to these resources was initially confined to batched runs that produced departmental reports at the appropriate intervals. A strong central information service department ruled the corporation. The role of the rest of the corporation limited to requesting new or more frequent reports and to provide hand written forms from which the central data banks were created and updated. The earliest client server solutions therefore could best be characterized as "SLAVE-MASTER". Time-sharing changed the picture. Remote terminal could view and even change the central data, subject to access permissions. And, as the central data banks evolved in to sophisticated relational database with non-programmer query languages, online users could formulate adhoc queries and produce local reports without adding to the MIS applications software backlog. However remote access was through dumb terminals, and the client server remained subordinate to the Slave\Master.

- Then, we provide a comprehensive taxonomy that covers key aspects of cloud-based data store: data model, data dispersion, data consistency, data transaction service, and data management cost.
- Finally, we map various cloud-based data stores projects to our proposed taxonomy to validate the taxonomy and identify areas for future research.

SURVEY 4

Publication: Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes.

Author: M. Tao, J. Zuo, Z. Liu, A. Castiglione

- The Smart Home concept, associated with the pervasiveness of network coverage and embedded computing technologies is assuming an ever-growing significance for people living in the highly developed areas.
- However, the heterogeneity of devices, services, communication protocols, standards and data formats involved in most of the available solutions developed by different vendors, is adversely affecting its widespread application.
- In this paper, promoted by several promising opportunities provided by the advances in Internet of Things (IoT) and Cloud Computing technologies for facing these challenges, a novel multi-layer cloud architectural model is developed to enable effective and seamless interactions/interoperations on heterogeneous devices/services provided by different vendors in IoT-based smart home.
- In addition, to better solve the heterogeneity issues in the presented layered cloud platform, ontology has been used as a promising way to address data representation, knowledge, and application heterogeneity, and an ontology-based security service framework is designed for supporting security and privacy preservation in the process of interactions/interoperations.
- A key benefit of connecting edge and cloud computing is the capability to achieve high-throughput under high concurrent accesses, mobility support, real-time processing guarantees, and data persistency.