International Journal On Engineering Technology and Sciences – IJETS™
ISSN(P): 2349-3968, ISSN (O): 2349-3976
Volume III, Issue V, May- 2016

PRIVACY PRESERVING WITH ANONYMOUS AUTHENTICATION FOR CLOUD COMPUTING ENVIRONMENT

Mr.S.Thiyagarajan
II-ME(CSE)
SSM College of Engineering, Komarapalayam,
Tamil Nadu, India
apachethiyagu@gmail.com

Mr.D.NAMACHIVAYAM M.E.,
Associate Professor CSE
Department of Computer Science and Engineering
SSM College of Engineering, Komarapalayam
Tamil Nadu, India

ABSTRACT

Cloud computing is an emerging data interactive paradigm to realize users data remotely stored in an online cloud server. Cloud services provide great conveniences for the users to enjoy the on-demand cloud applications without considering the local infrastructure limitations. During the data accessing, different users may be in a collaborative relationship, and thus data sharing becomes significant to achieve productive benefits. The existing security solutions mainly focus on the authentication to realize that a user's privative data cannot be illegally accessed, but neglect a subtle privacy issue during a user challenging the cloud server to request other users for data sharing. The challenged access request itself may reveal the user's privacy no matter whether or not it can obtain the data access permissions. In this paper, we propose a shared authority based privacy-preserving authentication protocol (SAPA) to address above privacy issue for cloud storage. In the SAPA shared access authority is achieved by anonymous access request matching mechanism with security and privacy considerations (e.g authentication, data anonymity, user privacy, and forward security);attribute based access control is adopted to realize that the user can only access its own data fields proxy re-encryption is applied to provide data sharing among the multiple users Meanwhile, universal composability (UC) model is established to prove that the SAPA theoretically has the design correctness. It indicates that the proposed protocol is attractive for multi-user collaborative cloud applications.

INTRODUCTION

CLOUD computing is a promising information technologyarchitecture for both enterprises and individuals. Itlaunches an attractive data storage and interactive paradigm with obvious advantages, including on-demand self services, ubiquitous network access, and location independent resource pooling [1]. Towards the cloud computing, a typical service architecture is anything as a service (XaaS), in which

infrastructures, platform, software, and others are applied for ubiquitous interconnections. Recent studies have been worked to promote the cloud computing evolve towards the internet of services [2], [3]. Subsequently, security and privacy issues are becoming key concerns with the increasing popularity of cloud services. Conventional security approaches mainly focus on the strong authentication to realize that a user can remotely access its own data in on demand mode. Along with the diversity of the application requirements, users may want to access and share each other's authorized data fields to achieve productive benefits, which brings new security and privacy challenges for the cloud storage.

An example is introduced to identify the main motivation. In the cloud storage based supply chain management, there are various interest groups (e.g., supplier, carrier, and retailer) in the system. Each group owns its users which are permitted to access the authorized data fields, and different users own relatively independent access authorities. It means that any two users from diverse groups should access different data fields of the same file. There into, a supplier

may want to access a carrier's data fields, but it is not sure whether the carrier will allow its access request. If the carrier refuses its request, the supplier's access desire will be revealed along with nothing obtained towards the desired data fields. Actually, the supplier may not send the access request or withdraw the unaccepted request in advance if it firmly knows that its request will be refused by the carrier. It is unreasonable to thoroughly disclose the supplier's private information without any privacy considerations. illustrates three revised cases to address

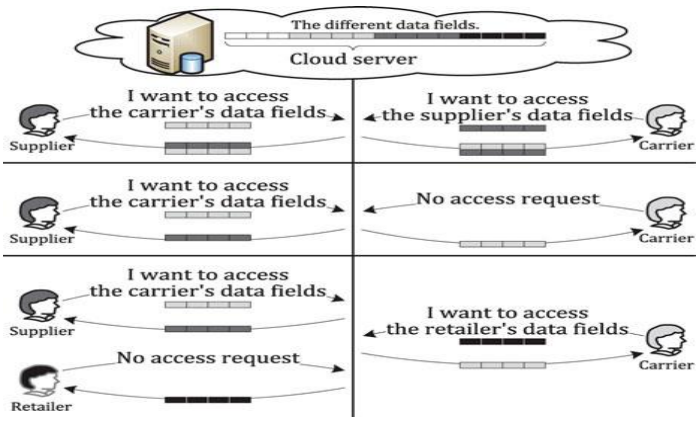
above imperceptible privacy issue.

_ Case 1. The carrier also wants to access the supplier's data fields, and the cloud server should inform each other and transmit the shared access authority to the both users;

_ Case 2. The carrier has no interest on other users' data fields, therefore its authorized data fields

should be properly protected, meanwhile the supplier's access request will also be concealed;

_ Case 3. The carrier may want to access the retailer's data fields, but it is not certain whether the retailer will accept its request or not. The retailer's authorized data fields should not be public if the retailer



SYNOPSIS

This project discusses the security issues involved in log management for a Secure Logging as a service and presents a design and implementation of a prototype delegating secure log manager. Main goal of a log manager is to provide high bandwidth and low level inactivity. In many real world applications and sensitive information must be kept in log files on an untreated machine. The event that an attacker captures this machine we would like to guarantee that he will gain little or no information from the log files and to limit his ability to corrupt the log file.

It describes a computationally cheap method for making all log entries generated prior to the logging machine's compromise impossible for the attacker to read and also impossible to undetectably modify or destroy. In this work, find out the challenges for a secure cloud based log management service. The attackers use below three steps to hack. First, the attacker can intercept any message sent over the Internet. Second, the attacker can synthesize, replicate, and replay messages in his possession. And Last The attacker can be a legitimate participant of the network or can try to impersonate legitimate hosts. We implement how to store secure log file in cloud and that file we can change read, write, delete, upload and download. We can implement AES algorithm that uses for log monitor and log generator.

EXISTING SYSTEM

Data handling in the cloud goes through a complex and dynamic hierarchical service chain. This does not exist in conventional environments. Ordinary web framework Uses web services for request and responses.

LIMITATIONS

- No security for user's data. No authentication or security provided
- High resource costs needed for the implementation.
- Not suitable for small and medium level storage users.

PROPOSED SYSTEM

In this paper, we propose a comprehensive solution for storing and maintaining log records in a server operating in a cloud-based environment. We address security and integrity issues not only just during the log generation phase, but also during other stages in the log management process, including log collection, transmission, storage, and retrieval. The major contributions of this paper are as follows. We propose architecture for the various components of the system and develop cryptographic protocols to address integrity and confidentiality issues with storing, maintaining, and querying log records at the honest but curious cloud

provider and in transit.

Literature Survey

1)Reliable Delivery and Filtering for Syslog

First Published: November 17, 2006 Last Updated: November 14, 2008

The Reliable Delivery and Filtering for Syslog feature allows a device to be customized for receipt of syslog messages. This feature provides reliable and secure delivery for syslog messages using Blocks Extensible Exchange Protocol (BEEP). Additionally, it allows multiple sessions to a single logging host, independent of the underlying transport method, and provides a filtering mechanism called a message discriminator. This module describes the functions of the Reliable Delivery and Filtering for Syslog feature and how to configure them in a network.

2)Guide to Computer Security Log Management

Recommendations of the National Institute of Standards and Technology

Karen Kent Murugiah Souppaya

It provides practical, real-world guidance on developing, implementing, and maintaining effective log management practices throughout an enterprise. The guidance in this publication covers several topics, including establishing log management infrastructures, and developing and performing robust log management processes throughout an organization. The publication presents log management technologies from a high-level viewpoint, and it is not a step-by-step guide to implementing or using log management technologies.

3)Explorative Visualization of Log Data to support Forensic Analysis and Signature Development

Sebastian Schmerl, Michael Vogel, René Rietz, and Hartmut König

Computer Networks and Communication Systems Group

Brandenburg University of Technology,Cottbus, Germany

In this paper, we propose an approach for log resp. audit data representation, which aims at simplifying the analysis process for the security officer. For this purpose audit data and existing relations between audit events are represented graphically in a three dimensional space. We describe a general approach for analyzing and exploring audit or log data in the context of this presentation paradigm. Further, we introduce our tool, which implements this approach and demonstrate the strengths and benefits of this presentation and exploration form.

4)On the Security of Public Key Protocols

DANNY DOLEV AND ANDREW c. YAO, MEMBER, IEEE

The Use of public key encryption to provide secure network communication has received considerable attention.Such public key systemsa

re usually very effective against a "passive" eavesdropper, namely, one who merely taps the communication line and tries to decipher the intercepted message. However, as pointed out in Needham and Schroeder an improperly designed protocol could be vulnerable to an "active" saboteur, one who may impersonate another user and may alter or replay the message. As a protocol might be compromised in a complex way, informal arguments that assert the security for a protocol are prone to errors

5)Architecture of an Open Object-Oriented Database Management System

David L. Wells, Jose A. Blakeley, and Craig W. Thompson Texas Instruments

An open, incrementally extensible object oriented database management system lets developers tailor database functionality for applications. It can also serve as a platform for research. This article describes the architecture of the Open OODB system. First we discuss its requirements, then its computational model. which builds database functionality as an extensible collection of transparent extensions to existing programming languages. We also describe how Open OODB's *system architecture* is decomposed into a kernel *meta-architecture* and a collection of modules implementing specific behavioral extensions. Finally, we discuss risks of the approach and report on the project's status.

6)Concurrency Control in Distributed Object-Oriented Database Systems

Kjetil Nørv°ag, Olav Sandst°a, and Kjell Bratbergsengen Department of Computer and Information Science, Norwegian University of Science and Technology In this paper we have given results from simulations with two different scheduler strategies. Further work for the DBsim simulator includes extensions that could make it more suitable for simulation of algorithms for object-oriented databases. Obviously, much more can be done with both the simulation model and the simulator. This includes adding new schedulers to the system, e.g., other versions of the two-phase locking scheduler, like wound-wait and wait-die. In a real system, replication is used for increased reliability and performance.

MODULES

- Log Generators
- Logging Client or Logging Relay
- Logging Cloud
- Log Monitor

Log Generators

These are the computing devices that generate log data. Each organization hat adopts the cloud-based log management service has a number oflog generators. Each of these generators is up to with logging capability. The log files generated by these hosts are not stored locally except temporarily till such time as they

are pushed to the logging client.

Logging Client or Logging Relay

The logging client is a collector that receives groups of log records generated by one or more log generators, and prepares the log data so that it can be pushed to the cloud for long term storage. The log data is transferred from the generators to the client in batches, either on a schedule, or as and when needed depending on the amount of log data waiting to be transferred. The logging client in corporates security protection on batches of accumulated log data and pushes each batch to the logging cloud. When the logging client pushes log data to the cloud it acts as a logging relay. We use the terms logging client and logging relay interchangeably. The logging client orrelay can be implemented as a group of collaborating hosts. For simplicity however, we assume that there is a single logging client.

Logging Cloud

The logging cloud provides long term storage and maintenance service to log data received from different logging clients belonging to different organizations. The logging cloud is maintained by a cloud service provider. Only those organizations that have subscribed to the logging cloud's services can upload data to the cloud. The cloud, on request from an organization can also delete log data and perform log rotation. Before the logging cloud will delete or rotate log data it needs a proof from the requester that the latter is authorized to make such a request. The logging client generates such a proof. However, the proof can be given by the logging client to any entity that it wants to authorize.

Log Monitor

These are hosts that are used to monitor and review log data. They can generate queries to retrieve log data from the cloud. Based on the log data retrieved, these monitors will perform further analysis as needed. They can also ask the log cloud to delete log data permanently, or rotate logs.

Algorithm:

Pushing or pulling strategies have interesting tradeoffs. The pushing strategy is beneficial when there are a large number of accesses to the data within a short period of time. In this case, if the data are not pushed out frequently enough, the log file may become very large, which may increase cost of operations like copying data. The pushing mode may be preferred by data owners who are organizations and need to keep track of the data usage consistently over time. For such data owners, receiving the logs automatically can lighten the load of the data analyzers. The maximum size at which logs are pushed out is a parameter which can be easily configured while creating the logger component. The pull strategy is most

needed when the data owner suspects some misuse of his data; the pull mode allows him to monitor the usage of his content immediately. A hybrid strategy can actually be implemented to benefit of the consistent information offered by pushing mode and the convenience of the pullmode.

5 FORMAL SECURITY ANALYSES WITH THE UNIVERSAL COMPOSABILITY MODEL

5.1 Preliminaries

The universal compos ability model specifies an approach for security proofs [28], and guarantees that the proofs will remain valid if the protocol is modularly composed with other protocols, and/or under arbitrary concurrent protocol executions. There is a real-world simulation, an ideal-world simulation, and a simulator Sim translating the protocol execution from the real-world to the ideal-world. Additionally, the Byzantine attack model is adopted for security analysis, and all the parties are modeled as probabilistic polynomial-time Turing machines (PPTs), and a PPT captures whatever is external to the protocol executions. The adversary controls message deliveries in all communication channels, and may perform malicious attacks (e.g., eavesdropping, forgery, and replay), and may also initiate new communications to interact with the legal parties. In the real-world, let p be a real protocol, Pi (i ¼ f1; . . . ;Ig 2 N_) be real parties, and A be a real-world adversary. In the ideal-world, let F be an ideal functionality, ~ Pi be dummy parties, and ~A be an ideal-world adversary. Z is anointer active environment, and communicates with all entities except the ideal functionality F. Ideal functionality acts as an un corrupt able trusted party to realize specific protocol functions. Theorem 1. UC Security. The probability, that Z distinguishes between an interaction of A with Pi and an interaction of ~A with ~ Pi, is at most negligible probability. We have that a real protocol p UC-realizes an ideal functionality F, i.e., IdealF; ~ A;Z Realp;A;Z. The UC formalization of the SAPA includes the idealworld model Ideal, and the real-world model Real. Ideal: Define two uncorrupted idea functionalities {Faccess, Fshare}, a dummy party ~ P (e.g., ~ Uu, ~ S,u 2 fa; bg), and an ideal adversary ~ A. { ~ P, ~ A} cannot establish direct communications. ~ A can arbitrarily interact with Z, and can corrupt any dummy party ~ P, but cannot modify the exchanged messages.

_ Real: Define a real protocol pshare (run by a party P including Uu and S) with a real adversary A and an environment Z. Each real parties can LIU ET AL.: SHARED AUTHORITY BASED PRIVACY-PRESERVING
AUTHENTICATION PROTOCOL IN CLOUD COMPUTING 247

communicate with each other, and A can fully control the interconnections of P to obtain/modify the exchanged messages. During the protocol execution, Z is activated first, and dual session identifiers shared by all the involved parties reflects the protocol state.

5.2 Ideal Functionality

Definition 1. Functionality Faccess. Faccess is an incorruptible ideal data accessing functionality via available channels, as In Faccess, a party P (e.g., Uu, S) is initialized (via input

Initialize), and thereby initiates a new session along with generating dual session identifiers {sidUu , sidSu }. P follows the assigned protocol procedure to send (via input Send) and receive (via input Receive) messages. A random number rPu is generated by P for further computation (via input Generate). Data access control is realized by checking {sendð:Þ, recð:Þ, localð:Þ} (via input Access). If P is controlled by an ideal adversary ~ A, four types of behaviors may be

performed: ~ A may record the exchanged messages on listened channels, and may forward the intercepted messages to P (via request Forward); ~ A may record the state of authentication between Uu and S to interfere in the normal verification (via request Accept); ~ A may impersonate an legal party to obtain the full state (via request Forge), and may replay the formerly intercepted messages to involve the ongoing communications (via request Replay).

Definition 2. Functionality Fshare. Fshare is an incorruptible ideal authority sharing functionality,

Fshare is activated by P (via input Activate), and the initialization is performed via Initialize of Faccess. The access request pointers {Rub Ua , RUa Ub } are respectively published and

challenged by {Ua, Ub} to indicate their desires (via input Challenge). The authority sharing between {Ua, Ub} is realized, and the desired data fields { _D Ub , _D Ua } are accordingly

obtained by {Ua, Ub} (via input Share). If P is controlled by an ideal adversary ~ A, ~ A may detect the exchanged challenged access request pointer RUx Uu(via request Listen); ~ A

may record the request pointer to interfere in the normal authority sharing between Ua and Ub (via request Forge/Replay).

In the UC model, Faccess and Fshare formally define the basic components of the ideal-world simulation. Party. Party P refers to multiple users Uu (e.g., Ua, Ub), and a cloud server S involved in a session. Through a successful session execution, {Uu, S} establish authentication and access control, and {Ua, Ub}

The session identifiers sidUu and sidSu are generated for initialization by the environment Z. The ideal adversary ~ A may control and corrupt the interactions between Uu

and S. Access request pointer. The access request pointer RUx Uu is applied to indicate Uu's access request on Ux's temp authorized data fields _D Ux .

5.3 Real Protocol pshare

A real protocol pshare is performed based on the ideal functionalities to realize Fshare in Faccess-hybrid model. Upon input ActivateðPÞ at P (e.g., Uu, and S), P is activated

Conclusion

This system proposed innovative approaches for automaticallylogging any access to the data in the cloud together withan auditing mechanism. Our approach allows the dataowner to not only audit his content but also enforce strongback-end protection if needed. Moreover, one of the mainfeatures of our work is that it enables the data owner toaudit even those copies of its data that were made withouthis knowledge.

Future work

In addition to a class file for authenticating the servers or the users, another class file finding the correct inner JAR, a third class file which checks the JVM's validity using oblivious hashing.

References:

[1] P. Ammann and S. Jajodia, "Distributed Timestamp Generation in Planar Lattice Networks," ACM Trans. Computer Systems, vol. 11, pp. 205-225, Aug. 1993.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. ACM Conf. Computer and Comm. Security, pp. 598- 609, 2007.

[3] E. Barka and A. Lakas, "Integrating Usage Control with SIP-Based Communications," J. Computer Systems, Networks, and Comm., vol. 2008, pp. 1-8, 2008.

[4] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology, pp. 213-229, 2001.

[5] R. Bose and J. Frew, "Lineage Retrieval for Scientific Data Processing: A Survey," ACM Computing Surveys, vol. 37, pp. 1- 28, Mar. 2005.

[6] P. Buneman, A. Chapman, and J. Cheney, "Provenance Management in Curated Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '06), pp. 539-550, 2006.

P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing," Nat'l Inst. of Standards and Technology, 2009.

[2] A. Mishra, R. Jain, and A. Durresi, "Cloud Computing: Networking and Communication Challenges," IEEE Comm. Magazine,
vol. 50, no. 9, pp. 24-25, Sept. 2012.

[3] R. Moreno-Vozmediano, R.S. Montero, and I.M. Llorente, "Key Challenges in Cloud Computing to Enable the Future Internet of Services,"IEEEInternet Computing,vol.17,no. 4,pp. 18-25,http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6203493, July/Aug.2013.

[4] K. Hwang and D. Li, "Trusted Cloud Computing with Secure Resources and Data Coloring," IEEE Internet Computing, vol. 14, no. 5, pp. 14-22, Sept./Oct. 2010.

[5] J. Chen, Y. Wang, and X. Wang, "On-Demand Security Architecture for Cloud Computing," Computer, vol. 45, no. 7, pp. 73-78, 2012.

[6] Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 12,pp. 2231-2244, Dec. 2012.

[7] H. Wang, "Proxy Provable Data Possession in Public Clouds,"IEEE Trans. Services Computing, vol. 6, no. 4, pp. 551-559, http:// ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6357181,Oct.-Dec. 2012.

[8] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 9, pp. 1717-1726, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6311398,Sept. 2013.

[9] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859-25, May 2011.

[10] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," IEEE Network, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[11] L.A. Dunning and R. Kresman, "Privacy Preserving Data Sharing with Anonymous ID Assignment," IEEE Trans. Information Forensics and Security, vol. 8, no. 2, pp. 402-413, Feb. 2013.

[12] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1182-1191, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6374615,June 2013.

[13] S. Grzonkowski and P.M. Corcoran, "Sharing Cloud Services: User Authentication for Social Enhancement of Home Networking," IEEE Trans. Consumer Electronics, vol. 57, no. 3, pp. 1424-1432, Aug. 2011.

[14] M. Nabeel, N. Shang, and E. Bertino, "Privacy Preserving Policy Based Content

Sharing in Public Clouds," IEEE Trans. Knowledge and Data Eng., vol. 25, no. 11, pp. 2602-2614, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6298891, Nov. 2013.

[15] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.-June 2012.

[16] S. Sundareswaran, A.C. Squicciarini, and D. Lin, "Ensuring Distributed Accountability for Data Sharing in the Cloud," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 4, pp. 556-568, July/Aug. 2012.

[17] Y. Tang, P.C. Lee, J.C.S. Lui, and R. Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 6, pp. 903-916, Nov./Dec. 2012.