Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 10, October 2014, pg.52 – 57

RESEARCH ARTICLE

A Novel Approach for Distributed Accountability of Data Sharing in a PHP Cloud

B.Sumitha[1], V.Venkateshwarlu[2]

[1]Pursuing M.Tech in CSE Department at JNTU Hyderabad, India

[2]Department of Computer Science, Kakatiya University, Warangal, Telangana, India

[1] sumitha.bodla@gmail.com, [2] veldevenkat@gmail.com

Abstract— Distributed computing and Cloud computing both enables highly scalable services that are easily consumed over the Internet on demand basis where the major feature of the distributed and cloud services is that users' data are usually processed remotely in unknown machines that users do not own or operate or have control on their data while enjoying the convenience brought by this new emerging technologies. But on same hand a user fears of losing control of his or her own data that can become a significant fear factor to the wide adoption of cloud services. To address this sort of problems we propose a novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in a PHP based cloud by proposing an object-cantered approach that enables enclosing our logging mechanism together with users' data and policies that are designed individually. We leverage the PHP programmable capabilities to both create a dynamic and travelling object which also ensures user data access by triggering authentication and automated logging techniques by performing extensive experimental study that demonstrate the efficiency and effectiveness of the proposed solution based on PHP.

Keywords— PHP, token, distributed computing, cloud computing, data sharing, accountability

I. INTRODUCTION

A PHP tag cloud computing presents a new way to supplement the current consumption and delivery model for IT services that are based on the Internet by provides dynamically scalable and often virtualized resources as a service over the Internet. There are a number of notable commercial and individual cloud computing services based on PHP tags that includes Amazon, Google, Microsoft, Yahoo, and Salesforce [1].

In a PHP cloud data comprises of alphabets, numbers, multimedia, audio waves, etc. where a services provider abstract cloud implementation from the users who no longer need to be experts of technology infrastructure in other words a users may not know the machines which actually process and host their data while enjoying the convenience brought by this novel technology but users also start worrying about losing control of their own data which may contain vital information too. The data processed on PHP clouds are often outsourced which may lead to a number of data leakage issues that may be related to accountability of handling personally identifiable information which is considered to be a significant barrier to the wide adoption of cloud services.

To provide solution space to the above mentioned problem space it is essential to provide an effective mechanism for users to monitor the usage of their data in the cloud based on the service level agreements that are made at the time they sign on for services in the cloud. The present access control approaches developed for closed domains such as databases and operating systems that operate on a centralized server in distributed environments is not suitable because:

- Data handling can be outsourced by the direct cloud service provider (DCSP) to other entities.
- Each entity in DCSP is capable of delegate notification to registered PHP tag.
- Entities are capable of ping or join and leave the cloud in a flexible manner.

To overcome the above problems in this paper we propose a novel approach by name Direct Cloud Information Accountability (DCIA) framework which is based on the notion of information accountability unlike privacy protection technologies that are built on the hide-it-or-lose-it perspective by focusing on information accountability by which the data usage is kept transparent and trackable by the registered user. Our proposed DCIA framework provides peer-to-peer accountability in a highly distributed fashion where one of the main innovative features of the DCIA framework lies in its ability of maintaining lightweight and powerful accountability that combines aspects based on PHP cloud access control, PHP cloud usage control and its authentication.

In DCIA a data owner can track not only whether or not the service-level agreements are being honored but also enforce access and usage control rules as needed we also develop two distinct modes for auditing: enque mode and dequeue mode. The enqueue mode refers to logs being periodically sent to the data owner or stakeholder while the dequeue mode refers to an alternative approach where the user or an authorized party can retrieve the logs as needed.

The design of the DCIA framework presents substantial challenges that includes uniquely identifying DCSPs by ensuring reliability of the log file by adapting to a highly decentralized infrastructure that addresses the issues related to leverage and extend the programmable capability of PHP files to automatically log the usage of the users data by any entity in the PHP cloud where a user will send the data along with any policies such as access control policies, logging policies, exception policies, dispute policies that they want to enforce on a PHP file to a cloud service provider.

Any access to the data will lead to a trigger that authenticates logging mechanism through a local PHP file which is considered to be a type of enforcement as "robust binding" because policies and the logging mechanism travel with the data and this robust binding exists even when copies of the PHP file is created by which a user will have control over his data at any location because the decentralized logging mechanism meets the dynamic nature of the PHP cloud but also imposes challenges on ensuring the integrity of the logging facility that is based on its ability to contact its central point whose access to its enclosed data will be denied.

In our paper we mainly concentrate on multimedia files such as images and videos which represent common content type such as text/multi for end users and organizations which are increasing by day that host their older data into the cloud as part of the storage services that is offered by the utility computing paradigm feature. An image or video often reveal social and personal habits of users or organizations and to handle such data our approach must be such that it can handle personal identifiable information provided they are stored as multimedia files that may contain image or video with some hidden text within it.

We tested our DCIA framework in a PHP cloud testbed and our experiments demonstrate the efficiency of PHP cloud and its scalability and granularity of our approach and also we have used Weka 6.0 to generate test results. We also provide detailed security analysis and discuss the reliability and strength of our proposed architecture which is base on various nontrivial attacks launched by malicious users.

II. RELATED WORK

Privacy and security are the two main issues in a PHP Cloud computing environment that is because the cloud users data and applications both reside for at least certain specified amount of time on the cloud cluster which is owned and maintained by a third party which is being outsourced and nobody knows why their personal information is requested or how it will be used or passed on to other parties so if accountability is introduced and every detail is stored regarding the data access of data sent to cloud in an encrypted format and the processing is done on the encrypted data.

The output of the processing is processed by the cloud privacy manager who provides limited features which does not guaranteed data protection to reveal the correct result. To overcome this problem we propose a layered architecture for addressing the peer-to-peer trust management and accountability problem in almost all present systems.

Self-defending objects are those objects which has an extension of the object-oriented programming paradigm that is implemented in a PHP cloud where the software objects offer sensitive functions or sensitive data are responsible for protecting those functions/data by relying on a centralized database to maintain the access records and due to which the items being protected are held as separate files which tend to prevent

privacy leakage from indexing which can be further integrated with the DCIA framework that is proposed in this work.

The Proof-Carrying authentication (PCA) framework includes a high order logic language that allows quantification over predicates and focuses on access control for web services by validating code rather than monitoring cloud content. And another approach is based on strongly coupling content with access control that uses Identity-Based Encryption (IBE) which is least reliable methodology since it does not bind the content with the proposed rules but using IBE we can make strong encrypted content and the log files in-terms of attaining protection against chosen plaintext and cipher text attacks.

Many researchers have proposed that we need to ensure that no one can add or remove entries in the middle of a data transmission chain without detection by which we can assure that data is correctly delivered to the receiver with data accountability to monitor the usage of the data and ensure that any access to the data is tracked in a distributed environment using technique such as auditing in the proposed extension of current access control mechanisms.

III. PROBLEM STATEMENT

Let us consider an example which is the problem statement of the paper. Mr. Ali is a publisher by profession who plans to sell his publications in the form of e-books by using the PHP Cloud Services for his business in the cloud he has proposed below mentioned software requirements based on which SRS document is prepared.

- His books must be downloaded only by users who have paid for her services.
- Potential buyers are allowed to view his books first before they make the payment to obtain the download right.
- Due to the nature of some of his works only users from certain countries can view or download some ebooks.
- For some of his publications users are allowed to only view them for a limited time so that the users cannot reproduce his work easily.
- All of his ebooks must be only readable that is user cannot print or copy whole or part of the ebook.
- In case any dispute arises with a client he wants to have all the access information of that client.
- He wants to ensure that the cloud service providers of PHP cloud do not share his data with other service providers so that the accountability provided for individual users can also be expected from the cloud service providers.

With the above specifications provided by Mr. Ali we need to identify the common requirements and develop several biz-rules to achieve data accountability in the cloud. A authentic user who subscribed to a certain cloud service usually needs to send his/her data as well as associated access control policies to the service provider once that data is received by the cloud service provider the service provider it will grant access permissions such as read or download on the ebook.

Using traditional access control mechanism once the access permissions are granted on the ebook it will be fully available at the service provider and one of the requirement of end user is to track the actual usage of the data here we aim to develop novel logging and auditing techniques which satisfy the following requirements:

- The logging should be a distributed or decentralized process for adapting the dynamic nature of the PHP cloud by using the concept of log files that should be tightly bounded with the corresponding data being controlled and require minimal infrastructural support from any server.
- Whenever a user tries to access his or her own data it should be correctly and automatically logged and need to integrate techniques to authenticate the entity which accesses the data then cross verify and record the actual operations on the data along with the time that the data have been accessed.
- A log file that stores all happenings on the PHP cloud must be reliable and tamper proof to avoid illegal insertions or deletion and modification by malicious parties or virus attacks and in such case the recovery mechanisms are also desirable to restore damaged log files that caused technical problems.
- In a continuous process all the log files should be sent back to their data owners to inform them of the current usage of their data and happenings on the access of data and major aspect is all the log files should be retrievable anytime by their data owners when needed regardless the location where the files are stored since it is a distributed environment.
- The proposed system should not intervene or monitor the ebook recipients systems or neither its MAC message authentication code address nor it should introduce heavy communication and computation overhead that may make end user to feel as a hectic job.

IV. IMPLEMENTATION AND EXECUTION

DCIA comprises components such as logger and the log harmonizer where a logger is the component which is strongly coupled with the user data which is downloaded when the data is accessed and is copied and the log harmonizer forms the central component which allows the user access to the log files.

The logger is strongly coupled with users data where the tasks automatically logs access to the data items that are in an encrypted format using a public key of the content owner and periodically sends it to a log harmonizer which may also be configured to ensure the access and usage control policies that associates with the data.

A logger along with error correction and detection it requires a very minimal support from the server available in PHP cloud for deployment since there will be a tight coupling established between data and logger and logger never requires to be installed as that of an applet in java so we are overcoming performance issue over applet usage.

Whereas the job of log harmonizer is controlling error correction and detection information that combines with the encryption and authentication by providing robust and reliable recovery mechanism when the client is not trusted to ensure trustworthiness we perform auditing which uses a randomly generated master key (combination of both decryption key for decrypting logs and the enqueue-and-dequeue key or strategy which enqueues or dequeues a log file based on "on demand requirement").

In our implementation we create more than one loggers for a single end user based on the policies being created at the time of PHP cloud deployment for the same set of data items in this case a log harmonizer will merge multiple log records by securing a log file from corruption by carrying itself with the out logging facility in addition to auditing which improves performance.

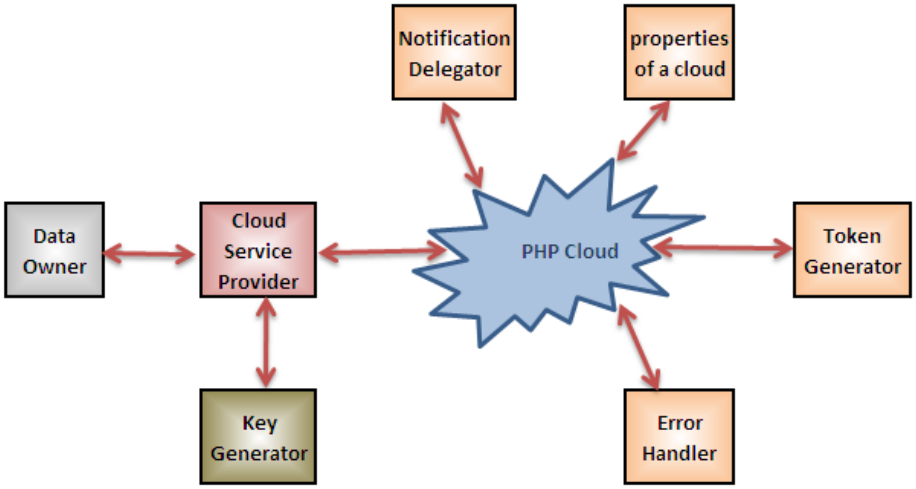


Fig. 1 Proposed PHP cloud information accountability framework.

When a user or admin tries to login and the authentication succeeds the responsibility of a service provider is to allow access with the data enclosed in the PHP cloud based on the configuration settings and policies defined at the time of deployment into PHP cloud which also provide usage control associated with logging or the logging functionality by activating the key generator which generates a public key to encrypt the data on cloud and is distributed by the data owner and stored at the cloud service provider as shown in the above figure fig1.

Token generator generates tokens to deploy on the cloud which are being requested by a data owner or a end user then in this process if any error is generated the error handler handles all types of errors and exceptions and when any such thing happens a notification delegators delegates notifications of happening in the form of a log file which is also sent by a logger. Our proposed framework is designed such a way that it dynamically prevents various attacks such as detecting illegal copies of user's data and encrypting data files to protect log files.

In a PHP cloud a entity is capable of performing read operation on data but is not allowed to save a raw copy of it anywhere permanently for performing this we use a PureLog tool which simply write a log record about the access while the access logs will enforce action through the access control module where the data is in an encrypted format and is capable of decrypting as and when required by the end user using PHP application when it is presented to a DCSP.

A raw copy of data is saved in an entity which is in a token format and does not possess any control over access of ebook copy or on the log records and by using PureLog the user's data can be directly downloadable in its original form using a hyper link and when user clicks on download button or hyper link a PHP file associated with the data that will decrypt the data and give it to the entity in raw form where as in case of AccessLogs the entire PHP file will be given to the entity as a token which is similar to a torrent file.

In enqueue mode a log file is periodically sent to the data owner or auditor by a PHP token harmonizer which leads to a trigger when a specified span of time elapses for a certain period according to the temporal timer inserted as part of the PHP SessonHandler file and one more trigger is fired when the expected file size which is predefined exceeds than that of intended at the time of creation. Once the log file is sent to the data owner the log files will be dumped for maintaining free space for future access logs which comprises of file access logs and exceptions raised information and enqueue to a user using PureLog tool.

The below are the result screens being generated based on our proposed system.

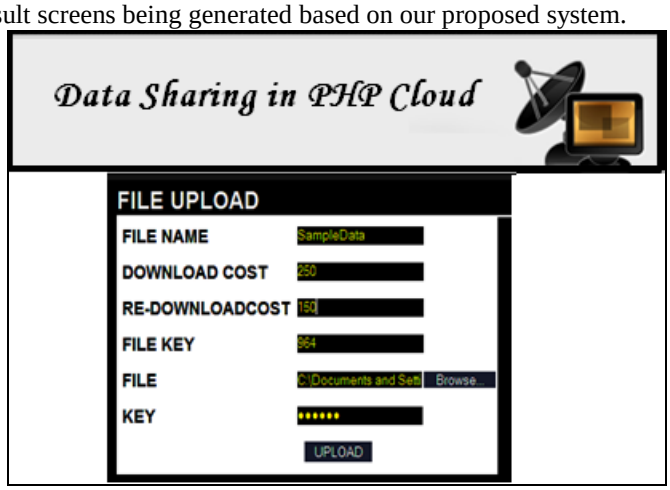


Fig. 2 File uploading screen in PHP Cloud

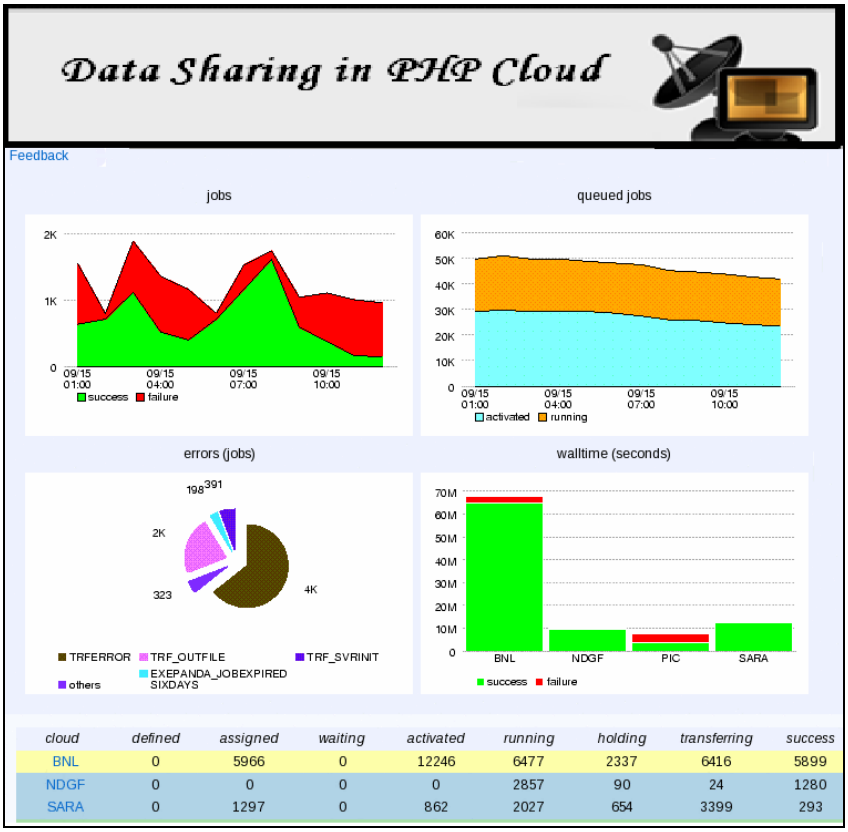


Fig. 3 PHP page depicting running jobs, queued jobs, error jobs and walltime in seconds implemented on different cloud service providers such as BNL, NDGF, SARA.

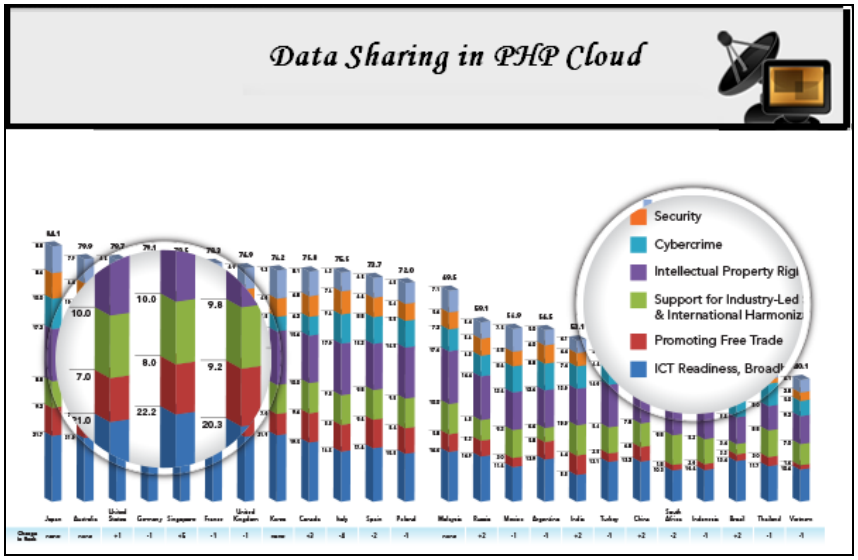
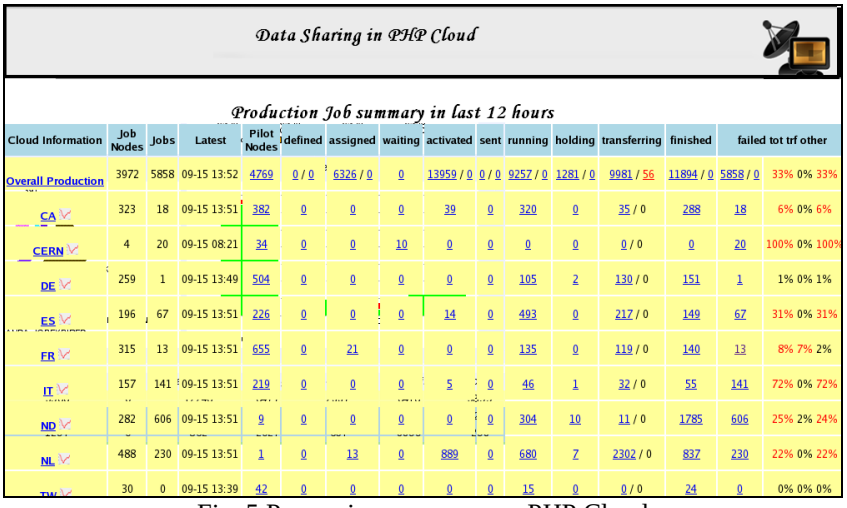


Fig. 4 PHP Cloud computing score card.



Cloud Information	Job Nodes	Jobs	Latest	Pilot Nodes	defined	assigned	waiting	activated	sent	running	holding	transferring	finished	failed	tot	trf	other
Overall Production	3972	5858	09-15 13:52	4769	0 / 0	6326 / 0	0	13959 / 0	0 / 0	9257 / 0	1281 / 0	9981 / 56	11894 / 0	5858 / 0	33%	0%	33%
CA	323	18	09-15 13:51	382	0	0	0	39	0	320	0	35 / 0	288	18	6%	0%	6%
CERN	4	20	09-15 08:21	34	0	0	10	0	0	0	0	0 / 0	0	20	100%	0%	100%
DE	259	1	09-15 13:49	504	0	0	0	0	0	105	2	130 / 0	151	1	1%	0%	1%
ES	196	67	09-15 13:51	226	0	0	0	14	0	493	0	217 / 0	149	67	31%	0%	31%
FR	315	13	09-15 13:51	655	0	21	0	0	0	135	0	119 / 0	140	13	8%	7%	2%
IT	157	141	09-15 13:51	219	0	0	0	5	0	46	1	32 / 0	55	141	72%	0%	72%
ND	282	606	09-15 13:51	9	0	0	0	0	0	304	10	11 / 0	1785	606	25%	2%	24%
NL	488	230	09-15 13:51	1	0	13	0	889	0	680	7	2302 / 0	837	230	22%	0%	22%
TW	30	0	09-15 13:39	42	0	0	0	0	0	15	0	0 / 0	24	0	0%	0%	0%

Fig. 5 Processing summary on PHP Cloud

V. CONCLUSION AND FUTUREWORK

In this paper we have proposed innovative approaches for automatically logging and accessing the data available or deployed on a PHP cloud that implements auditing mechanism where a data owner not only audits his ebooks but also enforce strong back-end protection on logs and the ebooks deployed and the main feature of our work is that it enables a data owner to audit even those copies of data that were made without his or her knowledge may be in the cloud. In the future, we plan to refine our approach to verify integrity and disk space management in the PHP cloud.

REFERENCES

[1] [1] P.T. Jaeger, J. Lin, and J.M. Grimes, "Cloud Computing and Information Policy: Computing in a Policy Cloud?," J. Information Technology and Politics,vol. 5, no. 3, pp. 269-283, 2009.

[2] [2] E. Barka and A. Lakas, "Integrating Usage Control with SIP-Based Communications," J. Computer Systems, Networks, and Comm., vol. 2008, pp. 1-8, 2008.

[3] [3] 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.

[4] [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] [5] P. Ammann and S. Jajodia, "Distributed Timestamp Generation in Planar Lattice Networks," ACM Trans. Computer Systems, vol. 11, pp. 205-225, Aug. 1993.

