Tracking of Data Leaving the Cloud

Yu Shyang Tan; Ryan K L Ko; Peter Jagadpramana; Chun Hui Suen; Markus Kirchberg; Teck Hooi Lim; Bu Sung Lee; Anurag Singla; Ken Mermoud; Doron Keller; Ha Duc

HP Laboratories
HPL-2012-108

Keyword(s):

Cloud Data Provenance; Cloud Security; Data Accountability

Abstract:

Data leakages out of cloud computing environments are fundamental cloud security concerns for both the end-users and the cloud service providers. A literature survey of the existing technologies revealed the inadequacies of current technologies and the need for a new methodology. This position paper discusses the requirements and proposes a novel auditing methodology that enables tracking of data transferred out of Clouds. Initial results from our prototypes are reported. This research is aligned to our vision that by providing transparency, accountability and audit trails for all data events within and out of the Cloud, trust and confidence can be instilled into the industry as users will get to know what exactly is going on with their data in and out of the Cloud.

External Posting Date: May 21, 2012 [Fulltext] Approved for External Publication
Internal Posting Date: May 21, 2012 [Fulltext]

Tracking of Data Leaving the Cloud

Yu Shyang Tan*, Ryan K L Ko*, Peter Jagadpramana*, Chun Hui Suen*, Markus Kirchberg*, Teck Hooi Lim*, Bu Sung Lee*, Anurag Singla†, Ken Mermoud†, Doron Keller†, Ha Duc†

*Hewlett-Packard Labs Singapore, Singapore
†ArcSight, HP Software, Cupertino, CA, USA

***Abstract*—Data leakages out of cloud computing environments are fundamental cloud security concerns for both the end-users and the cloud service providers. A literature survey of the existing technologies revealed the inadequacies of current technologies and the need for a new methodology. This position paper discusses the requirements and proposes a novel auditing methodology that enables tracking of data transferred out of Clouds. Initial results from our prototypes are reported. This research is aligned to our vision that by providing transparency, accountability and audit trails for all data events within and out of the Cloud, trust and confidence can be instilled into the industry as users will get to know what exactly is going on with their data in and out of the Cloud.**

***Keywords*-Cloud Data Provenance; Cloud Security; Data Accountability;**

I. Introduction

One of the top concerns of Cloud Computing is the handling of data in both public and private clouds. As Cloud Computing becomes mainstream, concerns regarding the security, storage and transfer of data within and out of cloud computing environments have only grown in importance [1]. To make matters worse, different countries have different data governance policies, e.g. one that does not permit company data to exit a specific boundary (such as a state or country). In a survey done by Fujitsu [2] in 2010, it was shown that 88 % of consumers are worried about who has access to their data and that some do not even want their data to be move out of the boundaries of their nation.

Therefore, the need for transparency and accountability of data in Cloud environments cannot be understated. We need mechanisms that can capture cloud data events and consolidate them into data-centric audit trails. Such audit trails will contain information on who, when, where and what actions have been performed on the data. As a result, users or administrators will have the ability to trace actions performed on the data for specific users or even track when their data has been taken out of the Cloud and by whom.

We believe that by providing such means to track their data both when inside and outside Cloud environments, confidence can be instilled among cloud consumers. To achieve full accountability, we take an end-to-end, inside-out approach. This means tracking the data both when it is within and when it is being taken out of the targeted cloud or distributed virtualized environment.

The challenges and approaches to building a framework for the derivation history of data (or data provenance) within a distributed virtualized environment is discussed in other works, mainly the TrustCloud project [3] and in [4].

Essentially, the idea is to utilize kernel-space data event logging mechanisms such as one described in [5] to capture events that are related to the data files. Using the captured information, security monitoring tools such as Arcsight [6] can than be used to detect and trace abnormal behaviours within the system. Having said so, the limitation here is that the data must reside within the said Cloud environment.

This paper serves two primary objectives. Firstly, it serves as a manifesto which highlights the research challenges and state-of-the-art of tracking data leaving cloud computing. Secondly, this paper introduces the early-stage results of our technique to track data which has been 'taken out' of a target cloud environment.

II. Use Cases

To illustrate the complexity and the research needs of tracking data in and out of the cloud from an end-to-end perspective, we describe two scenarios that can possibly happen when enterprises deploy part of their IT operations into the cloud. Even though the scenarios and possible solutions mentioned here are targeted at Cloud environments, they can also be applied to other forms of distributed environment such as shared office networks.

A. Use Case 1: Cloud Data Leakage

In a typical office setting, it is normal to have documents shared among multiple parties. These documents may also be classified as "highly sensitive" or "confidential". However, due to the need to share these sensitive files between multiple users, the risk of having the data document leaked out and not being able to track whose responsible is inevitably increased.

In a public cloud environment, such risk is even higher as the underlying hardware may possibly be shared between other users of the same public cloud (i.e. multi-tenancy). There is an inherent risk of having personnels who does not

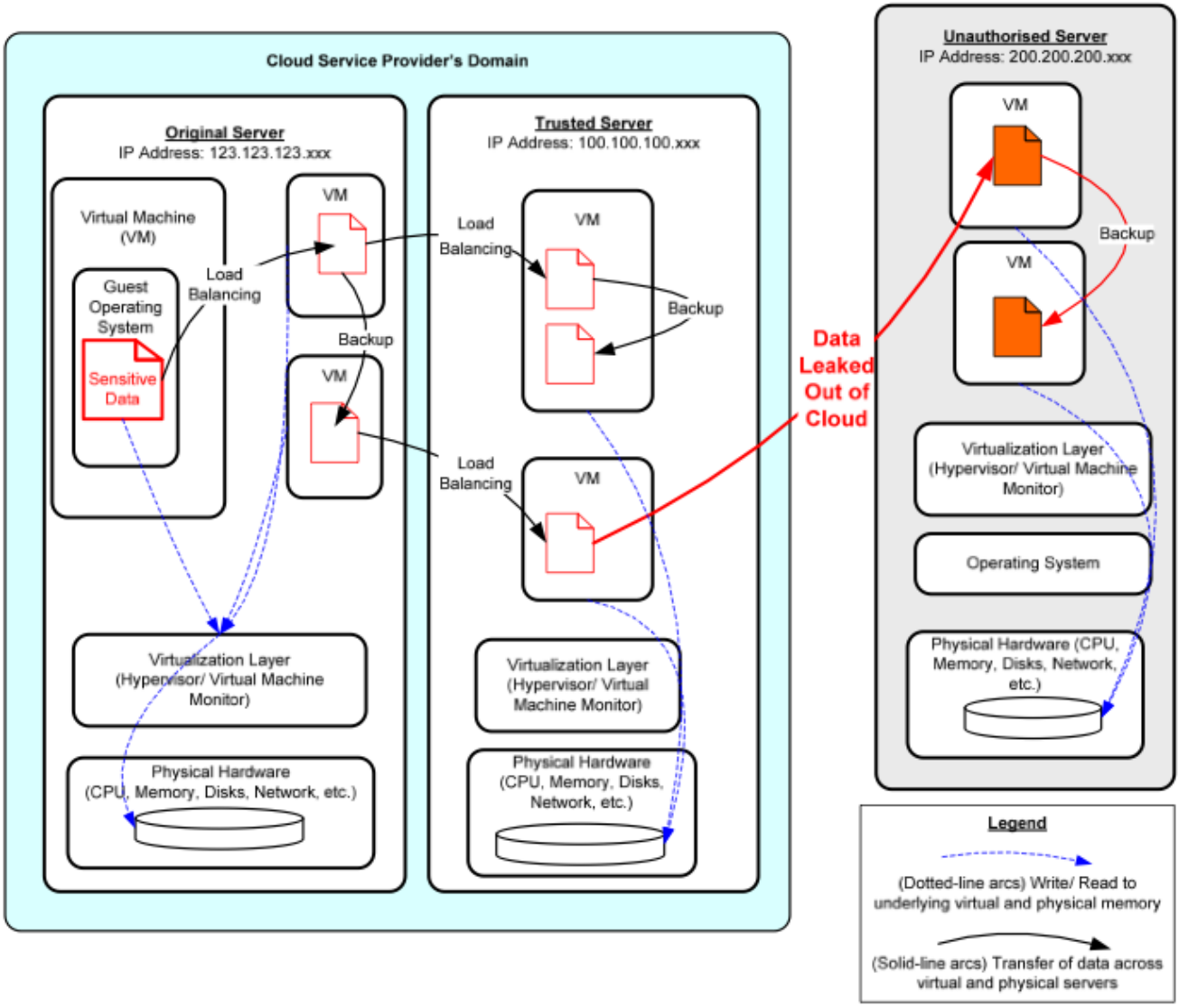

Figure 1. An illustration of the Cloud data leakage scenario

belong to the company to 'stumble upon' or hack into the underlying hardware and gain access to those documents.

In such situations, a malicious user, who can be a trusted employee of the company or someone from outside the company, can access the documents and copy them out of the public cloud storage space. Following which, the content of the documents can be renamed, modified and replaced back into the cloud storage in fraud cases, or the documents can be passed on to other external personnels who are interested in knowing the content of the documents. Often in either cases, it is very difficult for the authorities to trace who and how the information has leaked out, even if they know that there has been a leak. Figure 1 illustrates such a scenario.

However, with a mechanism that is capable of capturing events of the data even when it is being taken out of the said shared environment, tracing the person responsible for the leak can be easily achieved. Information such as who, when, where and actions performed can be logged as data events and sends the logs back to the owner of the data at regular intervals.

These data event logs can be compiled together to form the audit trails of the data. By looking at the provenance of the data, gathered from processing the data audit trails, the authorities will be able to quickly find out answers to questions such as who brought the data out of the cloud, where was it accessed outside the cloud or when was the data last modified. The processing of the data audit trails can even be done in near real-time, as the logs are being received at the server end, through the use of event monitoring tools or frameworks. Specific actions which can possibly lead to

a potential security breach, can be marked and flagged out by the monitoring tool, thus enabling the authorities to be alerted early and giving them time to execute damage control measures.

B. Use Case 2: Data Crossing Borders

Key companies such as banks often find themselves having to conform to laws and regulations pertaining to data, laid down by the country in which they operate in. These laws and regulations usually do not permit data documents from these companies to be stored outside of the country or state boundary for security and data sovereignty reasons.

With the current technologies, once the data documents have been taken out of the secured zone or in this case, the cloud environment, there is no way to know where the data has been taken to. This can be an issue in cases such as an ignorant employee who does not know that he is not suppose to bring the data out of the country or state borders or intended information theft.

With a more restrictive version of the event capturing mechanism, measures can be put into place to ensure that the policies and regulations are adhered to. We will describe this in more details when we present our approach in Section V.

For now, assuming that the mechanism is able to successfully capture events and sends them back to the owner's server, these events can be processed to find out who has accessed the data and the whereabouts of the data. Such information will be useful in tracking down the offender and apprehending him. From the preventive point of view, any attempts to access the data can be stopped by first getting the mechanism to seek permission to allow access from the main server. This is much like a user attempting to access a computer in a remote site and having to authenticate himself with an authentication server back at the main site. Except in this case, information such as who and where the data is, are being sent back to the server too. The accompanying event information will help the server in making the decision on whether to allow the user to access the data at the specified location.

III. Requirements of a Data Tracking Framework

While the need and usefulness of such data audit trails can be seen from the use cases mentioned, there is still the important question of how can these data audit trails be gathered and processed. A framework will be required to gather, store and process the incoming event logs before they can be analysed. The proposed framework should address the following issues:

- ***Tracking the data -*** The key component of the framework; the mechanism responsible for gathering the event logs should be able to capture all user actions performed on the data. In a controlled environment, such as in the Cloud, this can be achieved as the owner or administrator has control over the host machines used to access the data, i.e. System administrators can install monitoring software onto user machines. However, once out of the controlled environment, this is not possible. Constant monitoring of the data has to be maintained so as to avoid having "black out" periods where no user actions are being logged. This is to maintain consistency of the events being logged. Another factor that has to be taken note of is the granularity of the events being captured. Events should not be captured at a too abstract level as that will result in an audit trail which will be too general for forensic use. On the other hand, the events also cannot be captured at a too detailed level as that will result in an explosion in the size of the log files generated. This will cause problems to arise when storing and transmitting the log files. The granularity at which the events are captured should be at a level that will ensure the resulting data audit trails, constructed from the event logs, is able to provide a detailed enough timeline of user actions executed on the data such that when it is used in forensic work, it can allow tracing of specific actions to a specific user at any particular time or location.
- ***Sending of event logs back to the server -*** As stated previously, one of the major challenges is having the data being moved into an uncontrolled environment. As such, measures have to be put into place to ensure the consistency and integrity of the logs. Other than in the gathering phase, the integrity of the log files have to be tamper-proof when transmitting them back to the server for storage. Failure to do so will provide malicious users an easy avenue to defeat the intent of this system; to track and account for data that has been taken out of the boundaries of the restricted environment i.e. Cloud environment.
- ***Storing and processing of event logs -*** Once the data has been received at the server end, the logs have to be combined together with the provenance of the data file when it is still within the cloud environment to form the complete data audit trail for each data file. Considering the total size of the information to process is proportion to the number of data files in consideration and the level of activeness of each data file, a scalable approach is required to process these information. An efficient storage platform is required to provide a quick, efficient and scalable way to store and retrieve the composed audit trails. This phase is critical, especially if real-time monitoring of events is required, as the amount of delay in getting the audit trails ready will translate directly to the amount of time wasted in alerting the authorities when a security breach is suspected to have taken place.

IV. Related Works

In this section, we look at two of the most related fields, DRM/IRM and Watermarking techniques. We review the current technologies in each of these fields and evaluate their suitability and limitations for data accountability in the context of data leaving cloud computing environments.

A. DRM

Digital Rights Management (DRM) is a well known technology that is used for the management of digital contents. Most DRM related work in the academic looks at proposing a language model for authoring policies that governs the usage rights of the targeted digital content [7][8][9]. However, the main issue is the lack of sending of event logs back to the server and the eventual processing of the audit trails for data provenance.

Coincidentally, Diehl [10] did clearly state the need for provenance in the content protection layer, so as to allow the tracing of illegal documents back to the source. To achieve that, one has to know where, when and by whom was the copy being done.

The focus of this paper is rather on a framework that is able to provide data accountability even as the data is being moved out of the cloud and in the context of enterprises. The use cases and some of the requirements for enterprise DRM discussed by Arnab et al. in [11] pointed out the need for data provenance so as to enable tracking of the history of data.

On the industry side, several companies have rolled out implementations of DRM solutions for enterprises. Most notably are solutions such as Microsoft Windows Rights Management from Microsoft [12], Active Rights Management from Authentica [13], Oracle's IRM [14] and Apple's Fairplay [15]. While some of them have been discontinued due to controversy in user rights between rights holder and consumers, nevertheless, they are solutions which are related to our approach.

The main emphasis of these DRM solutions are similar; the security and access control of the target digital contents. Bulk of the frameworks put in place for these solutions are for the authentication of users, securing of documents and digital media contents, licensing and access control and rights control of users as discussed in [16] and [17]. Base on an analysis done by Michiels et al. [18], while it is possible to track the documents, the tracking services put in place are insufficient for accounting of data. Specifically, these tracking services mostly track and generate only statistical usage information such as the number of times the document is downloaded and the type of licenses issued. Information gathered from these tracking services generally do not tell who, when accessed a document and what is being done to the documents. These information are essential if we are to make a case on data accountability in the cloud.

In comparison, the methodology we propose here focuses on tracking the data document itself rather than the securing and rights management issues of documents. Tracking of the data document is done at a much detailed level as compared to enterprise DRM solutions, so as to enable backwards tracing from an occurring event to find out information such as who is responsible for the cause of the event, where and when did it happened and possible loopholes in the system. Gathering information of activities pertaining to the data is only the beginning. There is a need to process the gathered information and flag the authorities when suspected foul play is detected. Since DRM solutions generally focuses more on the security and access control aspect, such monitoring elements is not within current DRM solutions.

B. Watermarking

Another technology that we looked at is Watermarking. Commonly used for copyright protection, proof of ownership, transaction tracking and verification of digital contents, it is mainly applied in multimedia digital contents such as image and audio. It is generally done by "tagging" signals within the digital content with a watermark as described in [19][20][21]. That way by checking the watermark, one will be able to know from whom this digital copy belongs to and whether it is the original copy or not. Another use of watermarking is creating tamper-proof digital content. This technique is described in [22] where using fragile watermarking approach, a watermark is embedded in the discrete wavelet domain of the digital content by quantizing the corresponding coefficients. The advantage is that it allow content owners to detect whether any modifications have been made.

Although tracking of whether changes have been made is possible through the use of watermarking, but in the context of data accountability, it is unable to collect and relay back information on who and where was the alteration made. There is also no way to track the location of the data documents. As such, in terms of data accountability, it is insufficient to just utilize only watermarking. However, its ability to prove a means for tamper-proofing a digital content is promising. Application of the technique will add another level of assurance and means to validate whether a document has been tampered.

V. Our Approach

Clearly, current technologies are still inadequate in providing data accountability, particularly for data that has been taken out of the cloud or any secure environments. We propose CloudDT, a data accountability framework, for tracking data that have been taken out of the cloud. We first propose a method in which actions or events, such as access and modification, to data residing outside the cloud can be tracked, logged and sent back to the main server. Following which, we show how the captured information

can be combined with the event logs of the data, when it is still within the cloud, to form the data audit trail which details the complete provenance of the data from when it first enters the cloud.

A. Data Tracker

As mentioned in Section III, some key challenges in tracking data outside the cloud are; how can the consistent capturing of events be maintained in an uncontrolled environment and how can the integrity of the logs be maintained when sending them back to the server. In additional, one also has to consider how can the data remain track-able by the mechanism, *e.g.* prevent illegal duplication of the document.

To tackle the various challenges, we make use of archival techniques to archive the data into a self-executing container to form a medium that is able to follow the data for tracking purposes and to protect the data from direct accesses. This is done at the point when the data is about to be moved from the Cloud. When a user attempts to copy out or remove one or more data files out of the Cloud boundaries, from either the VMs or the storage medium in the Cloud, a ”check-out” via a provided web interface is enforced. The selected data files are encapsulated together with a viewer program, into a self-executing container. This self-executing container functions much like a self-extracting zip file where once executed, instead of un-archiving the content of the container, the self-executing container will execute the viewer program that resides within the container. Once the various data files and the viewer program are archived into the container, the container is encrypted to prevent users from having direct access to the data files. In this manner, only the viewer will have direct access to the data files since only the viewer will have the key, which is pre-programmed into it at the point of creating the container, to decrypt the encryption.

The viewer program is a small program which serves two main purpose. First, it acts as an interface through which users can interact with the data and perform actions such as view, modify or delete, on the data. Activities will all be monitored by the viewer. This is possible since all interactions with the data is routed through viewer program. By doing so, the viewer program forms sort of a semi-controlled environment. That way, the second purpose of the viewer program can be fulfilled, which is to capture and log down the events being captured. An overview of the Data Tracker is shown in Figure 2.

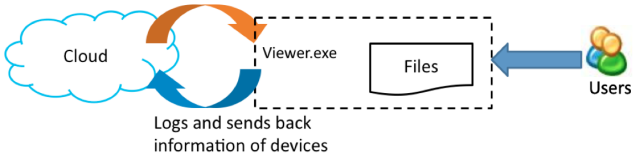

Figure 2. Overview of Data Tracker

After being invoked by the self-executing container, the viewer program will attempt to authenticate the user against a user credential list obtained from the authentication server in the cloud, when the data was being ”checked-out” initially. This is to establish the identity of the user accessing the data. A user session is created once the identity of the user has been successfully verified. Once done, the viewer program will begin monitoring the files and capture any events carried out by the user on the files. Events that are captured in each user session are linked to the user credential provided at the beginning. The granularity at which the events are being monitored, is set at the primitive file event level. Meaning to say the activities such as opening, modifying and closing of files are being monitored and captured except that the exact contents being modified are not monitored since that might result in the logs growing out of proportion. While the viewer program is authenticating the user, it also tries to retrieve information such as the IP address, host-machine name and other information that will allow the pinpointing of the location or specific identity of the host machine, from the underlying host Operating System (OS). Figure 3 shows an example of the types of information field that are being collected by the viewer program. These information regarding the host machine together with the user-linked events, are than stored into a temporary log file inside the container.

Having successfully logged all data events, the next step is to send these log files back to the server for further processing. This is done at the end of each user session. When a user session is closed, the viewer program will attempt to establish a connection with the main server back in the cloud domain. If successful, the viewer program will encrypt the files so as to ensure the integrity of the logs and sends the cipher back to the server. Standard AES and RSA encryption algorithms are used in the encryption process. In the event where the connection with the main server cannot be established, *e.g.* no internet connection, the viewer will delay sending the log files until the end of the next user session.

At this junction, the viewer allows a user to view and modify the data files inside the container. However, the viewer can be extended to support user-access control. This can be achieve by first retrieving a user access-rights list from the authentication server when the data is being checked out of the cloud. Together with the user-access list, the two lists are encapsulated into the container, together with the viewer program so as to allow the viewer access to user credentials and access-right information without having to contact the main authentication server.

B. Event Detection

The key focus of our proposed framework is to allow owners of the data files or the administrators to be able to monitor the data even when it does not reside within the

```
Timestamp,Timestamp_precision,Filename,Actions,User_name,Process_Id,Host_name,Host_IP,Host,Mac

2011-03-23 14:00:26,1300860025,mounts,Read,alice,1427,Na@hp,210.21.12.37,D0:66:83:FE:00:76:00:DF
2011-03-23 14:00:34,1300860033,readme.txt,Read,alice,1428,Na@hp,210.21.12.37,D0:66:83:FE:00:76:00:DF
2011-03-23 14:00:41,1300860040,mounts,Write,alice,1427,Na@hp,210.21.12.37,D0:66:83:FE:00:76:00:DF
2011-03-23 14:00:53,1300860053,readme.txt,Write,alice,1428,Na@hp,210.21.12.37,D0:66:83:FE:00:76:00:DF
2011-03-23 14:01:01,1300860060,document.txt,Read,alice,3029,Na@hp,210.21.12.37,D0:66:83:FE:00:76:00:DF
2011-03-23 14:01:14,1300860073,document.txt,Write,alice,3029,Na@hp,210.21.12.37,D0:66:83:FE:00:76:00:DF
2011-03-23 14:01:24,1300860083,document.txt,Close,alice,3029,Na@hp,210.21.12.37,D0:66:83:FE:00:76:00:DF
```

Figure 3. Sample of the information being collected by the viewer program

cloud. To reiterate the purpose; the ability to monitor the data itself will facilitate companies and owners of the data in the enforcement of data policies and regulations and assist them in the detection and tracing of malicious acts performed on the data itself. Gathering information on what is being done to the data, at where and by whom, when the data is not within the cloud is just the first stept. The next step is to treat the log information received such that it can be used to fulfil the above mentioned monitoring purposes.

When the server first receives the log files from the trackers, the information in the logs are stored into a database so as to prevent the loss of information. This is essential for scalability and ease of retrieval reasons. In the event when there are large amount of trackers sending back information, the logs have to be stored quickly and in a manner which allows ease in the retrieval of specific selections of the log files. Highly scalable distributed databases such as Vertica [23] can be used here as they can alleviate the worry of running out of storage space on a single server, mainly the receiving server while at same time allows retrieving sub portions of the log files. This will make things convenient for the next step; the generation of the data audit trail. Generation of the data audit trail is done by merging the log information with the event information gathered while the data is within the cloud's boundaries. The merge is done base on the file name followed by the timestamp that is tagged to each event. This is where the ability to retrieve file related events only instead of having to process iteratively each received log files, can benefit the generation of the audit trail. While selective retrieval of information from the log files can greatly reduce the data footprint, the challenge of having to process large amount of data still exists due to the fact that multiple files may be required to be processed at the same time.

Once the data audit trails are generated, rule-base monitoring systems such as Arcsight[6] and Drools [24] can be used to monitor the audit trails and violations such as unauthorized access to specific set of documents or accessing of the documents out of the permitted zone can be flagged out and the authorities notified immediately. Having said so, during our experiments on monitoring the data audit trails that we generated, we found that there are still some features lacking in the monitoring frameworks that we have tried using. Frameworks used for monitoring data audit trails should be able to trace back the ancestry of an event. Tracing the ancestry of an event will enable the creation of process trees, effectively showing the chain of processes that leads up to the violation. Such process trees is useful in post event detective efforts, *e.g.* tracing back who and where did the violation happen or where are the security loopholes that have allowed such violations. To overcome the inability to track the ancestry of events, we build a visualization interface to map the events together. An example of the resulting visualization is shown in Figure 4.

The visual component is used mainly for manually tracing where the fault lies within the process tree. Through analyzing such process tree visualizations, one can easily answer questions such as; who and when secretly retrieve a document out from the cloud and where was the document accessed outside the cloud and what was being done to the document. With the aid of the visualization and the data audit trail, data in the cloud can now be accounted for. Detective approaches can be used to find out who, where and what has been done to the data.

VI. Limitations and Future Work

We recognize possible weaknesses in the data tracker component of our proposed solution. Due to the need to operate in an uncontrolled environment, it is possible that the data tracker can be compromised and as such, the data can be taken out of the viewer. By doing so, the framework that we proposed here will not be able to continue tracking the compromised data and as such the data is deemed "lost". Addressing this potential limitation is our current top priority. As future work, we plan to look into using Trusted Computing to ensure that the remote machine on which the data tracker is to operate on, is attestable. Lamacchia [25] and Erickson [26] both discussed on the use of Trusted Computing for ensuring that the policies set out in DRM is enforced correctly. Trusted Computing might provide an alternative solution to ensuring that the data tracker is not being compromised. That said, we strongly believe that our proposed technique is the start of a new era for tracking data out of cloud computing environments.

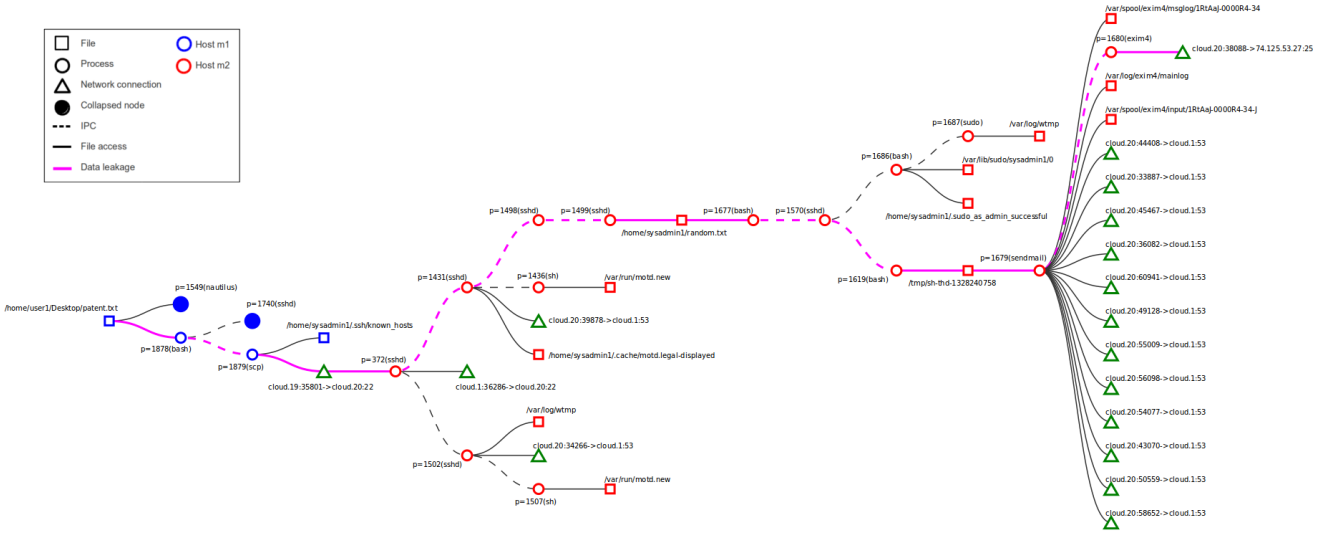

Figure 4. Visualization showing the process tree of a flagged event

VII. Concluding Remarks

We believe that in order for Cloud to be more readily acceptable by the industry, there must be a means where data can be accounted for. We reviewed some of the current technologies used in the tracking of data usage and pointed out the inadequacies of those technologies to provide data accountability for data that have been moved out of the cloud.

As a initial step, we proposed a methodology, CloudDT, in which data can be tracked even when it is being moved out of the cloud. Combined with the work done in [3],[4] and [5], we show how by using monitoring tools, suspected violations of policies or regulations can be noticed and flagged.

Detective work in finding out details on where and when the violation has taken place, who is responsible and possible security loopholes in the system, can be carried out by analyzing the data audit trails gathered.

On an ending note, it is our utmost hope that our initial work serve as a catalyst for the beginning of vested research efforts in this impending cloud computing security problem.

VIII. Acknowledgements

The authors would like to thank all members of HP Labs Singapore's TrustCloud research project team. At the same time, the authors would like to thank the HP ArcSight team for their in-depth discussions, critique and integration efforts.

References

[1] K. Kwang, "Cloud to come of age by 2015," Web. [Online]. Available: http://www.zdnetasia.com/cloud-to-come-of-age-by-2015-62302900.htm

[2] "Personal data on the cloud: A global survey of customer attitudes," 2010. [Online]. Available: http://www.fujitsu.com/global/news/publications/dataprivacy.html

[3] R. K. L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "Trustcloud: A framework for accountability and trust in cloud computing," in *IEEE World Congress on Services*, 2011.

[4] O. Q. Zhang, M. Kirchberg, R. K. L. Ko, and B.-S. Lee, "How to track your data: The case for cloud computing provenance," in *CloudCom*, 2011, pp. 446–453.

[5] R. K. L. Ko, P. Jagadpramana, and B. S. Lee, "Flogger: A file-centric logger for monitoring file access and transfers within cloud computing environments," in *IEEE TrustCom/IEEE ICESS/FCST, International Joint Conference of.* Los Alamitos, CA, USA: IEEE Computer Society, 2011, pp. 765–771.

[6] "Arcsight." [Online]. Available: http://www.arcsight.com/products/, 2011

[7] V. Rosset, C. V. Filippin, and C. M. Westphall, "A drm architecture to distribute and protect digital contents using digital licenses," in *Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop*, ser. AICT-SAPIR-ELETE '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 422–427. [Online]. Available: http://dx.doi.org/10.1109/AICT.2005.5

[8] P. A. Jamkhedkar and G. L. Heileman, "A formal conceptual model for rights," in *Proceedings of the 8th ACM workshop on Digital rights management*, ser. DRM '08. New York, NY, USA: ACM, 2008, pp. 29–38. [Online]. Available: http://doi.acm.org/10.1145/1456520.1456528

[9] A. Arnab and A. Hutchison, "Persistent access control: a formal model for drm," in *Proceedings of the 2007 ACM*

workshop on Digital Rights Management, ser. DRM '07. New York, NY, USA: ACM, 2007, pp. 41–53. [Online]. Available: http://doi.acm.org/10.1145/1314276.1314286

[10] E. Diehl, "A four-layer model for security of digital rights management," in *Proceedings of the 8th ACM workshop on Digital rights management*, ser. DRM '08. New York, NY, USA: ACM, 2008, pp. 19–28. [Online]. Available: http://doi.acm.org/10.1145/1456520.1456527

[11] A. Arnab and A. Hutchison, "Requirement analysis of enterprise drm systems," in *In Proceedings of Information Security South Africa (ISSA) Conference 2005, Johannesburg, South Africa*, 2005.

[12] D. Shinder, "How the windows rights management service can enhance the security of your documents." [Online]. Available: http://www.windowsecurity.com/articles/Windows%5FRights%5FManagement%5FService%5FDocuments.html

[13] "Authentica introduces e-drm solution for secure information collaboration." [Online]. Available: http://www.econtentmag.com/Articles/News/News-Item/Authentica-Introduces-E-DRM-Solution-for-Secure-Information-Collaboration-6120.htm

[14] "Oracle irm." [Online]. Available: http://www.argentra.com/cms4/products/oracle/oracle-irm.html

[15] "How fairplay works: Apple's itunes drm dilemma." [Online]. Available: http://www.roughlydrafted.com/RD/RDM.Tech.Q1.07/2A351C60-A4E5-4764-A083-FF8610E66A46.html

[16] V.-V. Patriciu, I. Bica, M. Togan, and S.-V. Ghita, "A generalized drm architectural framework," *Advances in Electrical and Computer Engineering*, vol. 11, pp. 43–48, 2011.

[17] R. Iannella, "Digital rights management architectures." [Online]. Available: http://www.dlib.org/dlib/june01/iannella/06iannella.html

[18] S. Michiels, K. Verslype, W. Joosen, and B. De Decker, "Towards a software architecture for drm," in *Proceedings of the 5th ACM workshop on Digital rights management*, ser. DRM '05. New York, NY, USA: ACM, 2005, pp. 65–74. [Online]. Available: http://doi.acm.org/10.1145/1102546.1102559

[19] C. Podilchuk and E. Delp, "Digital watermarking: algorithms and applications," *Signal Processing Magazine, IEEE*, vol. 18, no. 4, pp. 33 –46, jul 2001.

[20] I. Cox and M. Miller, "Electronic watermarking: the first 50 years," in *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, 2001, pp. 225 –230.

[21] B. Macq, J. Dittmann, and E. Delp, "Benchmarking of image watermarking algorithms for digital rights management," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 971 – 984, june 2004.

[22] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1167 –1180, jul 1999.

[23] "Vertica." [Online]. Available: http://www.vertica.com/

[24] "Drools." [Online]. Available: http://www.jboss.org/drools

[25] B. A. Lamacchia, "Key challenges in drm: An industry perspective," in *Proceedings of the 2nd ACM DRM Workshop (in conjunction with ACM CCS Conference*, 2002.

[26] J. S. Erickson, "Fair use, drm, and trusted computing," *Commun. ACM*, vol. 46, no. 4, pp. 34–39, Apr. 2003. [Online]. Available: http://doi.acm.org/10.1145/641205.641228