Cloudopsy: an Autopsy of Data Flows in the Cloud

Angeliki Zavou[1], Vasilis Pappas[1], Vasileios P. Kemerlis[1], Michalis Polychronakis[1], Georgios Portokalidis[2], and Angelos D. Keromytis[1]

[1] Columbia University, New York, NY, USA
{azavou,vpappas,vpk,mikepo,angelos}@cs.columbia.edu
[2] Stevens Institute of Technology, Hoboken, NJ, USA
gportoka@stevens.edu

Abstract. Despite the apparent advantages of cloud computing, the fear of unauthorized exposure of sensitive user data [3, 4, 8, 13] and non-compliance to privacy restrictions impedes its adoption for security-sensitive tasks. For the common setting in which the cloud infrastructure provider and the online service provider are different, end users have to trust the efforts of both of these parties for properly handling their private data as intended. To address this challenge, in this work, we take a step towards elevating the confidence of users for the safety of their cloud-resident data by introducing Cloudopsy, a service with the goal to provide a visual *autopsy* of the exchange of user data in the cloud premises. Cloudopsy offers a user-friendly interface to the customers of the cloud-hosted services to *independently* monitor and get a better understanding of the handling of their cloud-resident sensitive data by the third-party cloud-hosted services. While the framework is targeted mostly towards the end users, Cloudopsy provides also the service providers with an additional layer of protection against illegitimate data flows, *e.g.,* inadvertent data leaks, by offering a graphical more meaningful representation of the overall service dependencies and the relationships with third-parties outside the cloud premises, as they derive from the collected audit logs. The novelty of Cloudopsy lies in the fact that it leverages the power of *visualization* when presenting the final audit information to the end users (and the service providers), which adds significant benefits to the understanding of rich but ever-increasing audit trails. One of the most obvious benefits of the resulting visualization is the ability to better understand ongoing events, detect anomalies, and reduce decision latency, which can be particularly valuable in real-time environments.

1 Introduction

The benefits of cloud computing for service providers and end users have led to a rapid increase of online services and applications. As businesses and individuals rely on the cloud for most of their everyday tasks, it is inevitable that *personally identifiable information* (PII) will also be stored and processed on third parties premises, like the cloud, outside the administrative control of its owners. Credit card numbers, private files, and other kinds of sensitive data are temporarily or permanently stored in back-end databases and file systems, beyond user control. In this setting, data confidentiality becomes one of the primary concerns for users of these cloud-hosted services, especially when taking into account the recent security incidents and data breaches [1,5,15,16], witnessed even on major and reputable online services. Customer lack of confidence is actually one of the most highly cited concerns regarding the use of cloud-hosted services [7]. In lack of an alternative option (other than not using the service at all), most users eventually share their data with cloud services, and rely on legal agreements and trust the efforts of service providers in securely handling and protecting their data.

However, even when service providers are considered trusted and follow best security practices, unauthorized access to sensitive data remains a plausible threat, *e.g.,* due to software vulnerabilities, improper access permissions to third-party services, misconfigurations, or incorrect assumptions. Although, there has been a large body of work on the prevention, detection, and mitigation of such incidents, they still pose an tangible threat. From the data owner's perspective, users wish to have a better understanding on how their data is being handled by the cloud-hosted online services, and ensure that their data is not being abused or leaked, or at least have an indication that such an event has occurred.

Auditing has been long employed with success as an added security measure in alleviating user security concerns in domains like banking, where information security is mission critical. Unlike other privacy protection technologies that are built on enforcing policies and preventing data flows, auditing focuses on keeping data usage *transparent* and *trackable*. Thorough, efficient auditing in cloud computing remains a challenge even for straightforward web services, but more research is directed towards this goal [2, 10, 12, 18]. However, even when auditing mechanisms are implemented successfully, services continuously increasing in size (*i.e.,* number of users, components, *etc.*), create massive numbers of audit logs and trails of information. In consequence, their usefulness is limited because the task of interpreting logs and identifying interesting details becomes extremely challenging for end users. Therefore, it is easy to conclude that the audit information collected must be structured in a way that is comprehensible by end users.

To address this privacy challenge, we take a step towards increasing cloud transparency with respect to the handling of sensitive user data by introducing Cloudopsy, a service with the goal of providing *secure* and *comprehensible* data auditing capabilities to both the service providers and their clients for user data collected in the realm of these cloud-hosted services. Even though information flow auditing and enforcement techniques use similar mechanisms, we focus our efforts in *auditing*, due to the concerns that enforcement could have adverse effects like breaking parts of services. Cloudopsy operates on together with our cloud-wide data tracking framework [14] and it enhances with the power of *visualization*, when presenting the final audit information to data owners and service providers. This novel feature of Cloudopsy significantly reinforces end-

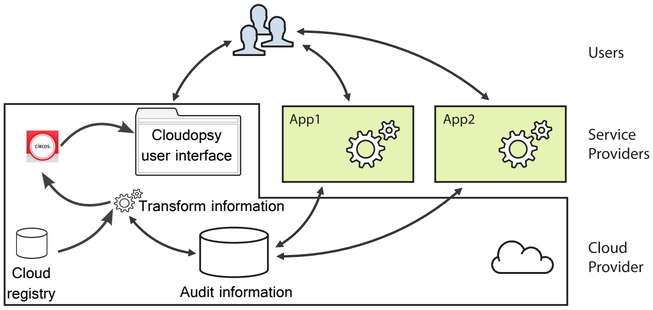

Fig. 1. High-level overview of Cloudopsy's architecture and how it integrates with our cloud-wide data tracking framework.

user awareness regarding the use and exposure of their sensitive data by the cloud-hosted applications. In a few words, we claim that audit visualization is a necessary feature in the analysis of audit logs, as it helps in recognizing events and associations recorded in the logs that might have otherwise remained unnoticed or would have taken longer to realize.

To truly support this vision, we envision cloud providers offering the Cloudopsy auditing service in addition to their existing hosting environment, to both service providers and data owners. A large, reputable cloud provider can help leverage user confidence much more effectively. Cloudopsy platform architecture could dramatically reduce the per-application development effort required to offer data protection, while still allowing rapid development and maintenance.

The remainder of this paper is organized as follows: Section 2 discusses our goals and key assumptions. We present Cloudopsy and elaborate on its components in sections 3 and 4. In Section 4, we demonstrate Cloudopsy with a real-world application and conclude the paper in Section 5.

2 Cloudopsy

A high-level overview of Cloudopsy's architecture is illustrated in Fig. 1. The participating entities are: the *cloud provider*, who offers the cloud infrastructure or platform, the *service providers* deploying the online service, and the *end users* or *data owners*, who are the clients of the online service. Cloudopsy is an enhanced auditing service offered by the cloud provider to the service providers it hosts on its premises, as well as the end users of these services.

The providers of the cloud-hosted services need to define the boundaries of their *data flow domain*, *i.e.,* the component applications of their *composite* service (*e.g.,* web server, back-end database *etc.*). On the other hand, the users of the cloud-hosted services are assigned a *unique ID*, which will be their unique cloud-wide identification. Each end user has to register/sign in with this ID every time he uses one of the services hosted by the cloud provider, if he wishes to have his sensitive data audited for the services hosted on the same cloud provider. (*e.g.,* Google Apps).

The rest of Cloudopsy's mechanism can be divided in three main components: (a) the generation of the audit trails, (b) the transformation of the audit logs for efficient processing, and (c) the visualization of the resulting audit trails.

2.1 Audit Logs Generation

To generate audit logs, Cloudopsy relies on information flow tracking techniques to tracks the entire flow of information in the realm of participating services hosted on the same cloud provider [14]. Separate audit logs are created for each cloud-hosted service and they are later correlated and filtered for creating the different audit logs that will be presented to data owner or the service provider requesting the audit. From a high level view, after one user's data enter the service, they are colored with a unique ID bound to that particular user, and audit information is generated every time it crosses the defined boundary of the data flow domain of the service, *e.g.,* when colored data is about to be send to a cloud storage device or host inside or outside of the cloud, that does not belong to the defined components of the service. These events are recorded in an audit back-end database in an "append-only" fashion. Each component of the online service hosted on the cloud provider premises pushes audit messages to the audit database. Note that the auditing mechanism of Cloudopsy is designed to generate "verbose" and detailed audit logs. Although the same audit trails will be used to provide audit information to both service providers and end users, the final information displayed to them will be different. The end user sees information regarding his own data, whereas the service provider has access to the audit logs of all user data handled by his service.

In addition to identifying individual user's data, the colors that are assigned to user data can also reflect their type. That can be accomplished by assigning them a second ID or sacrificing some bits of information, previously part of the user ID, and using them for indicating their type or class. For instance, possible classes of data could refer to credit card numbers, e-mails, social security numbers (SSN), *etc.* The class of user data can be assigned by the service provider, or would be automatically determined by content scanners [6] that identify known patterns of PII like SSNs.

2.2 Audit Trails Processing

The audit logs produced from the auditing component include raw log data in a textual form and need further processing before reaching the visualization component. The necessary transformations include: (a) mapping of the applications that comprise the cloud-hosted services, (b) filtering of the information from the "verbose" audit logs and transforming them to a format understood by the visualization component, and (c) correlating relevant information stored in several log files and generated by different cloud applications exchanging user data.

More specifically, Cloudopsy maintains a global registry of the active connections for the cloud-wide domain that it monitors. In order to transform the audit logs in an appropriate format for the next phase, technical details are used from this registry, and through automated procedures combined to create a more appropriate presentation of the collected audit information for input to the visualization component. Several other mechanisms are applied to the "verbose" audit logs to extract only the necessary segments for final log information that will be presented to the end users. For example, information unrelated to "suspicious" flows of user data will not be extracted for the next phase.

2.3 Audit Trails Visualization

The visualization mechanism of Cloudopsy is a significant enhancement to previous auditing mechanisms as the power of images will offer to both service providers and end users, a better understanding as well as deeper insights in the real flows of user data. It remains a challenge though to choose the appropriate and meaningful graphical representation of the collected audit data, which we discuss in more details in Sec. 4. In a few words, the visualization component receives the transformed audit logs and process them according to a set of different parameters and represents the graphic results in a circular layout representing the cloud on Cloudopsy's web interface. The services monitored by Cloudopsy are included in the final graphical representation and the recorded transfers of the colored user data will be represented by links between them as well, colored and directed according to the information from the audit logs. As expected, the generated output from the visualization component differs not only between users but also between the data owners and the providers of the online service.

3 Cloud-wide auditing mechanism

In previous work we implemented a cloud-wide fine-grained data flow tracking framework [14] for cloud-hosted services. This DFT framework is used for the auditing of the data flows and the generation of the raw audit logs that will be analyzed by Cloudopsy. We assume that the service providers have integrated this mechanism in their applications to enhance the security of the provided services and leverage the generic facility offered by the cloud provider.

From a high-level perspective, the auditing mechanism is built on top of a user-level data flow tracking framework libdft [9], based on runtime instrumentation and is integrated into the components of the service and works as follows: the data that is "tagged" as *sensitive* is tracked across all local files, host-wide IPC mechanisms, and selected network sockets. Audit information selected by the auditing component is kept in a back-end database located outside the vicinity of the service providers, and more importantly operates in an append-only fashion for preventing tampering of the archived audit trails. The audit information collected by the DFT component of captures leakage events that result from writing the marked data into files or remote endpoints. Each entry of the audit log will include information about the time, the running application, the action, the destination stream (a network address or a file path) and the tagged data.

Our first prototype of the auditing component is implemented using Intel's binary instrumentation tool Pin 2.10, and works with unmodified applications running on x86-64 Linux. It performs byte-level data flow tracking and supports 32-bits tags, allowing support for 2^{32} different tags values (colors) per byte – consequently, the same number of different users. In a newest version of the auditing mechanism we used the last two digits of the tag to represent different classes of user's sensitive data. The auditing component provides a transparent, fine-grained and domain-wide data flow tracking mechanism suitable for our target cloud environment.

4 Visualization

Our decision on enhancing our audit log analysis mechanism with visualization techniques was influenced by user studies comparing the effectiveness of visual presentations, in comparison to linear textual presentations of similar results to the ones we are handling in our audit logs analysis. In particular, in our case, since we wanted to represent the relationships and flow patterns of user data within the services in and outside of the cloud, Circos [11] seemed as a good candidate. Although Circos was originally developed to provide a better display of the chromosomal relationship between various species, it has been successfully used also for the graphical representations of other types of data, *i.e.,* network traffic. We chose Circos also due to its support for a circular data domain layout (resembles the cloud) and its multiple customizations opportunities, and finally due to its competence of smoothly partitioning the final image into any number of disjoint regions, drawing attention to multiple different events in a single display. Circos uses a circular ideogram layout to facilitate the display of relationships between pairs of services running on the cloud by the use of links (straight lines or Bezier curves), which can visually encode the position, size, and orientation of interacting elements.

Circos requires a specific format for the input data, which we generate through the transformations described in Sec. 2. In addition to our transformed audit log files, we also created configuration files including the necessary parameters for correctly generating the output required. The goal was to effectively present, in the final output, the monitored data flows within the cloud, as well as the ones "leaking" information out of the cloud premises, faithfully matching what is described in the initial audit files.

We provide two types of visual representations of data flows. For end users, we present a graphical display of the flow of their sensitive data that the cloud-hosted online service obtained as the result of their interactions with this service, *e.g.,* a purchase from an online store would result into the storage of a user's credit card number in the service's back-end database. Cloudopsy in this case summarizes the transfers of this data in a circular graph, representing each data

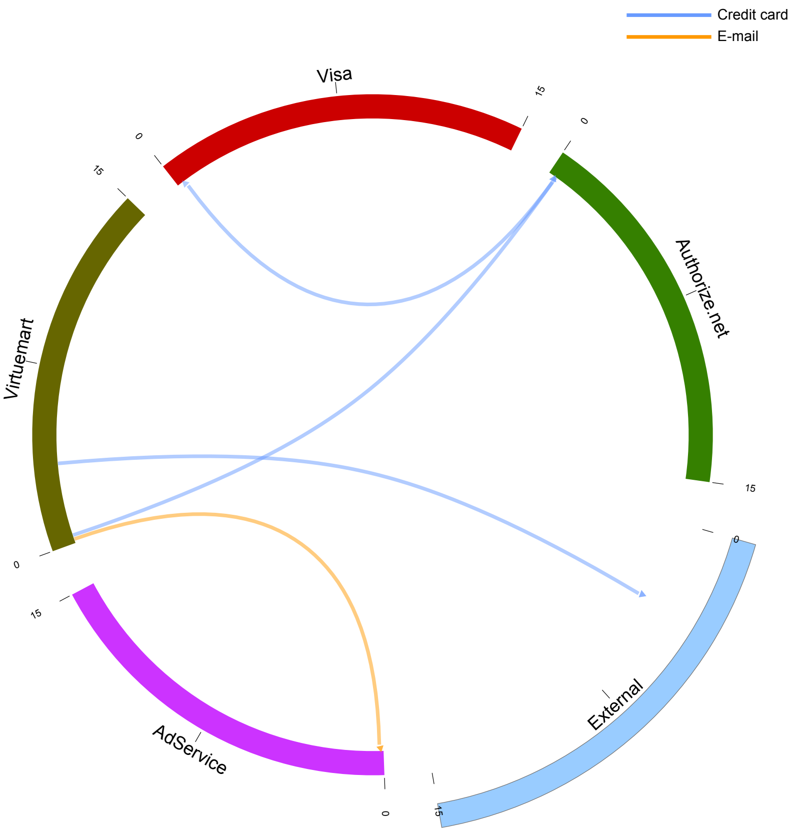

Fig. 2. A user's view of a transaction.

exchange as a link between the involved services. This can help users gain a deeper understanding of how their data are handled in a facile way.

On the other hand, Cloudopsy's output for the provider of the service is a much "richer" graph of the audited transmissions entailing information for *all* clients' sensitive data collected. In particular, since the service provider needs to analyze the very large and usually complex audit log files, the visualization of the audited events is extremely beneficial for him as he can immediately verify legitimate operations or identify unexpected "suspicious" transmission patterns that might have otherwise remained opaque in the reams of the audit logs. Furthermore, the service provider can also see the flow of data between components of his service that could help him identify internal errors.

Sample results of this process can be seen in figures 2 and 3. Figure 2 displays the movement of a single user's "marked" sensitive data, *i.e.,* credit card number and email address, as they move in time between the audit-enabled applications. On the other hand, Fig. 3, which is aimed for the provider of the service, is a much "richer" image showing at a first glance patterns in the data

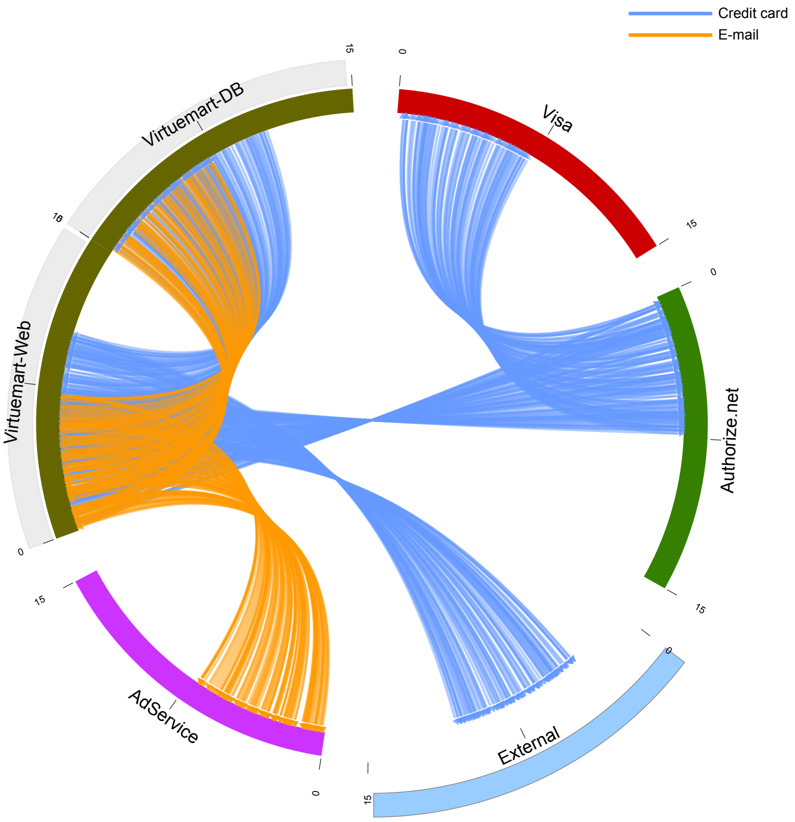

Fig. 3. A service provider's view of user transactions.

flows. In general, the final audit images presented on Cloudopsy's web interface include the communicating applications of the composite cloud-hosted services running over the Cloudopsy mechanism, as well as hosts outside the cloud which according to the logs participated in sensitive data transmissions. In general, the final circular graphs consist of an outer ring representing the cloud-hosted applications (colored blocks) running over the auditing infrastructure, whereas the colored arcs depict the recorded transmissions of monitored user data. More specifically, in these two figures the blue and orange colored directed links represent different classes of sensitive data. In particular, in Fig. 3, the different colors could also represent the different users of the service. It is on the provider's discretion to customize the parameters on Cloudopsy's web interface, in respect to the information he is interested in on different occasions.

Scenario: Testing Cloudopsy with an E-store

To test Cloudopsy we chose an e-store application, namely VirtueMart [17], hosted on a cloud-based infrastructure. We configured VirtueMart to accept only payments with credit card, and set up actual electronic payments through the Authorize.Net service using test accounts. Typically during a purchase transaction, users enter their personal data, credit card, email *etc.*, through the web front end of the application, which then are transmitted to the back-end database of the e-store, and to the payment processor. After their credit card is verified, the e-store approves the purchase and the transaction completes successfully. The personal data that the user enters remain stored (as is frequently the case) in the database of the e-store.

Figure 2 depicts the movement of the credit card number for one user that completed a successful purchase through VirtueMart. It also represents the email of this user also stored in the backend database of the e-store. Apart from the exchanged data though, this figure shows also the time that each event took place. (For each segment time progresses clockwise). Therefore, we can immediately identify two events. First, that the email is sent to the AdService. This might be a legitimate action in case the client did not opt out of sending his information for advertising reasons while using the online service. But the most interesting event presented in this figure is that credit card number seems to transfer to an unknown IP outside the cloud premises at a later point in time, which definitely looks suspicious and should be further investigated as a possible inadvertent data leak. Note that this cannot be the legitimate channel of the user, as this was known from the start of the transaction and should be whitelisted by the auditing mechanism as a legitimate flow and therefore would not be part of the audit logs. Even with this simple figure this unexpected event was very easily identified.

5 Conclusion

In this paper, we argue that the concerns of the end users of cloud-hosted services about the usage of their data can be a major deterrent for users looking to embrace cloud-hosted services. We focused our efforts on enhancing the effectiveness of the information collected in the audit log files and presented Cloudopsy, a framework that through visualization and automated analysis of the results and based on the graphs produced with the Circos visualization tool, remediates cloud users security concerns and enables even users without any particular technical background to get a better understanding of the treatment of their data by third-part cloud-hosted services.

Acknowledgements This work was supported by DARPA and the National Science Foundation through Contract FA8650-11-C-7190 and Grant CNS-12-28748, respectively, with additional support from Google. Any opinions, findings, conclusions or recommendations expressed herein are those of the authors, and do not necessarily reflect those of the US Government, DARPA, NSF or Google.

References

1. Berghel, H.: Identity theft and financial fraud: Some strangeness in the proportions. Computer 45(1), 86–89 (jan 2012)

2. Chen, Y.Y., Jamkhedkar, P.A., Lee, R.B.: A software-hardware architecture for self-protecting data. In: Proceedings of the 2012 ACM conference on Computer and communications security. pp. 14–27. CCS '12 (2012)
3. Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., Molina, J.: Controlling data in the cloud: outsourcing computation without outsourcing control. In: Proceedings of the 2009 ACM workshop on Cloud computing security. pp. 85–90. CCSW '09 (2009)
4. Cloud Security Alliance: Security guidance for critical areas of focus in cloud computing v2.1 (December 2009), https://cloudsecurityalliance.org/csaguide.pdf
5. Computerworld: Microsoft BPOS cloud service hit with data breach (Dec 2010), http://www.computerworld.com/s/article/9202078/Microsoft_BPOS_cloud_service_hit_with_data_breach
6. Cornell University: Open-source Forensics Tools for Network and System Administrators – Spider. http://www2.cit.cornell.edu/security/tools/ (February 2010)
7. Gens, F.: IT Cloud Services User Survey, pt.2: Top Benefits & Challenges. IDC (October 2008), http://blogs.idc.com/ie/?p=210
8. Kaufman, L.: Data security in the world of cloud computing. Security Privacy, IEEE 7(4), 61 –64 (July 2009)
9. Kemerlis, V.P., Portokalidis, G., Jee, K., Keromytis, A.D.: libdft: Practical Dynamic Data Flow Tracking for Commodity Systems. In: Proc. of VEE (2012)
10. Ko, R., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., Lee, B.S.: TrustCloud: A framework for accountability and trust in cloud computing. In: Services (SERVICES), 2011 IEEE World Congress on. pp. 584–588 (july 2011)
11. Krzywinski, M.I., Schein, J.E., Birol, I., Connors, J., Gascoyne, R., Horsman, D., Jones, S.J., Marra, M.A.: Circos: An information aesthetic for comparative genomics. Genome Research (2009)
12. Massonet, P., Naqvi, S., Ponsard, C., Latanicki, J., Rochwerger, B., Villari, M.: A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In: Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on. pp. 1510–1517 (may 2011)
13. Molnar, D., Schechter, S.: Self hosting vs . cloud hosting : Accounting for the security impact of hosting in the cloud. In: Proceedings of the 9^{th} Workshop on the Economics of Information Security. pp. 1–18. WEIS '10 (2010)
14. Pappas, V., Kemerlis, V., Zavou, A., Polychronakis, M., Keromytis, A.D.: CloudFence: Enabling Users to Audit the Use of their Cloud-Resident Data. Tech. Rep. CUCS-002-12, CS Department, Columbia University (2012), http://hdl.handle.net/10022/AC:P:12821
15. Sophos: Groupon subsidiary leaks 300k logins, fixes fail, fails again (2011 Jun), http://nakedsecurity.sophos.com/2011/06/30/groupon-subsidiary-leaks-300k-logins-fixes- fail-fails-again/
16. The Wall Street Journal: Google Discloses Privacy Glitch (2009 Mar), http://blogs.wsj.com/digits/2009/03/08/1214/
17. VirtueMart eCommerce Solution: VirtueMart shopping cart software, http://virtuemart.net
18. Wang, C., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for data storage security in cloud computing. In: INFOCOM, 2010 Proceedings IEEE (march 2010)