International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 6th Mar 2018. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-03

Title: SECURING OUTSOURCING DATA IN THE CLOUD ENVIRONMENT

Volume 08, Issue 03, Pages: 46–51.

Paper Authors

MR.A.JANARDHAN RAO, P.UMASHANKAR

Vignan's Lara Institute of Technology & Science

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

SECURING OUTSOURCING DATA IN THE CLOUD ENVIRONMENT

MR.A.JANARDHAN RAO[1], P.UMASHANKAR[2]
Assistant Professor[1], Department of M.C.A, Vignan's Lara Institute of Technology & Science
M.C.A Student[2],Department of M.C.A, Vignan's Lara Institute of Technology & Science

Abstract:
Cloud storage platforms promise a convenient way for users to share files and engage in collaborations, yet they require all files to have a single owner who unilaterally makes access control decisions. Existing clouds are, thus, agnostic to the notion of shared ownership. This can be a significant limitation in many collaborations because, for example, one owner can delete files and revoke access without consulting the other collaborators. In this paper, we first formally define a notion of shared ownership within a file access control model. We then propose two possible instantiations of our proposed shared ownership model. Our first solution, called Commune, relies on secure file dispersal and collusion resistant secret sharing to ensure that all access grants in the cloud require the support of an agreed threshold of owners. As such, Commune can be used in existing clouds without modifications to the platforms. Our second solution, dubbed Comrade, leverages the block chain technology in order to reach consensus on access control decision. Unlike Commune, Comrade requires that the cloud is able to translate access control decisions that reach consensus in the block chain into storage access control rules, thus requiring minor modifications to existing clouds. We analyze the security of our proposals and compare/evaluate their performance through implementations using Amazon S3.

Introduction
Even though the cloud promises a convenient way for users to share files and effortlessly engage in collaborations, it still retains the notion of individual file ownership. That is, each file stored inthe cloud is owned by a single user, who can unilaterally decide whether to grant or deny any access request to that file. However, the individual ownership is not suitable for numerous cloud-based applications and collaborations. Consider a scenario where a number of research organizations and industrial partners want to set up a shared cloud repository to collaborate on a joint research project. If all participants contribute their research efforts to the project, then they may want to share the ownership over the collaboration files so that all access decisions are agreed upon among the owners. There are two main arguments why this may be preferred to individual ownership. First, a sole owner can abuse his rights by unilaterally making access control decisions. The community features a number of anecdotes where users revoke access to shared files from other collaborators. Second, even if owners are willing to elect and trust one of them to make access control

decisions, the elected owner may not want to be held accountable for collecting and correctly evaluating other owners' policies. For example, incorrect evaluations may incur negative reputation or financial penalties. In contrast to individual ownership, we introduce a novel notion of shared ownership where n users jointly own a file and each file access request must be granted by a pre-arranged threshold of t owners. We remark that existing cloud platforms, such as Amazon S3 or Drop box, provide no support for shared ownership policies, and offer only basic access control lists. In short, they are agnostic to the concept of shared ownership. Furthermore, state-of-the-art trust management systems that can support shared ownership policies (e.g., SecPAL, KeyNote, Delegation Logic) make all access decisions using a centralized Policy Decision Point (PDP). This is not suitable for enforcing our shared ownership model, because the user who administrates the PDP can arbitrarily change the policy rules set by the owners and enforce his own policies. In this paper, we address the problem of distributed enforcement of shared ownership within cloud storage providers. By distributed enforcement, wemean enforcement where access to files in a shared repository is granted if and only if t out of n owners separately support the grant decision. Therefore, we introduce the Shared- Ownership file access control Model (SOM) to define our notion of shared ownership, and to formally state the given enforcement problem. We then propose two instantiations of the SOM model to enforce shared ownership policies in a distributed fashion. This paper extends our previous work. More specifically, we provide additional formal details about the SOM model. We also propose a new instantiation of the SOM model, Comrade, that leverages functionality from the block chain in order to reach consensus on access control decisions. Unlike the Commune framework proposed, Comrade requires cooperation from the cloud provider that is expected to translate access control decisions that reached consensus in the block chain into storage access control rules. Comrade, however, exhibits considerably better performance than Commune. We deploy a smart contract instantiating Comrade within the Ethereum block chain, connect it to Amazon cloud storage [5], and compare its performance to the one of Commune with respect to the file size and the number of users. We summarize our contributions as follows: We formalize the notion of shared ownership within a file access control model named SOM, and use it to define a novel access control problem of distributed enforcement of shared ownership in existing clouds. We propose a first solution, called Commune, which distributively enforces SOM and can be deployed in an agnostic cloud platform. Commune ensures that (i) a user cannot read a file from a shared repository unless that user is granted read access by at least t of the owners, and (ii) a user cannot write a file to a shared repository unless that user is granted write access by at least t of the owners.

We propose a second solution, dubbed Comrade, which leverages functionality from the block chain technology in order to reach consensus on access control decision. Comrade improves the performance of Commune, but requires that the cloud is able to translate access control decisions that reached consensus in the block chain into storage access control rules, thus requiring minor modifications of existing clouds.

We build prototypes of Commune and Comrade and evaluate their performance within Amazon S3 with respect to the file size and the number of users.

Existing system:

The cloud promises a convenient way for users to share files and effortlessly engage in collaborations, it still retains the notion of individual file ownership. That is, each file stored in the cloud is owned by a single user, who can unilaterally decide whether to grant or deny any access request to that file. However, the individual ownership is not suitable for numerous cloud-based applications and collaborations. Consider a scenario where a number of research organizations and industrial partners want to set up a shared cloud repository to collaborate on a joint research project. If all participants contribute their research efforts to the project, then they may want to share the ownership over the collaboration files so that all access decisions are agreed upon among the owners. There are two main arguments why this may be preferred to individual ownership. First, a sole owner can abuse his rights by unilaterally making access control decisions. The community features a number of anecdotes where users revoke access to shared files from other collaborators. Second, even if owners are willing to elect and trust one of them to make access control decisions, the elected owner may not want to be held accountable for collecting and correctly evaluating other owners' policies.

Drawbacks:

1. Use a centralized repository owned by a single user because the repository owner can unilaterally grant or deny access to the files stored therein.

2. No Distributed enforcement of shared ownership in clouds.

Proposed system

More specifically, we provide additional formal details about the SOM model. We also propose a new instantiation of the SOM model, Comrade, that leverages functionality from the blockchain in order to reach consensus on access control decisions. Unlike the Commune framework proposed, Comrade requires cooperation from the cloud provider that is expected to translate access control decisions that reached consensus in the blockchain into storage access control rules. Comrade, however, exhibits considerably better performance than Commune. We deploy a smart contract instantiating Comrade within the Ethereum blockchain, connect it to Amazon cloud storage, and compare its performance to theone of Commune with respect to the file size and the number of users

Advantages:

Formalize the notion of shared ownership within a file access control model named

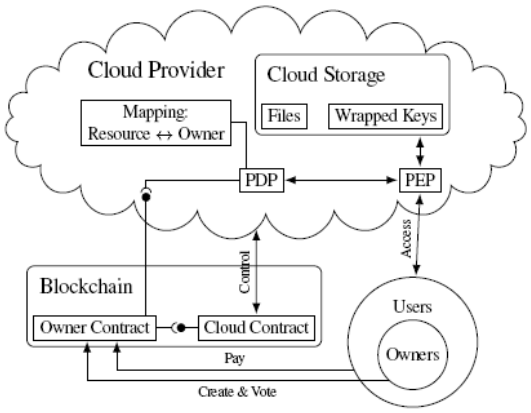
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

SOM, and use it to define a novel access control problem of distributed enforcement of shared ownership in existing clouds.

We propose a first solution, called Commune, which distributively enforces SOM and can be deployed in an agnostic cloud platform.

Comrade, which leverages functionality from the blockchain technology in order to reach consensus on access control decision.

Architecture



Modules:

SOM: SHARED-OWNERSHIP FILE ACCESS CONTROL MODEL

The concept of shared ownership, and formally instantiate it in a file access control model dubbed SOM. Our main motivation for constructing this model is three-fold: (i) to precisely define the ideal set of features that we believe a model, which enforces shared ownership, should provide; (ii) to formulate the problem of distributed enforcement more precisely by focusing on SOM's formal description; and (iii) to provide a point of reference to scrutinize SOM's enforcement solutions, including our own.

COMMUNE: DISTRIBUTED ENFORCEMENT OF SHARED OWNERSHIP IN AN AGNOSTIC CLOUD

This section presents Commune, our solution for distributed enforcement of the SOM access control policy in an agnostic cloud. As SOM does not specify concrete file access operations, we instantiate Commune with write and read actions. Commune cannot use a centralized repository owned by a single user because the repository owner can unilaterally grant or deny access to the files stored therein. Our alternative is to use a "shared repository", which is an abstraction built on top of the owners' personal accounts on S.

COMRADE: BLOCKCHAIN-BASED SHARED OWNERSHIP

Comrade with write and read actions. Before introducing our solution, we provide some background on the blockchain and describe the system mode.

Conclusion:

Even though existing cloud platforms are used as shared repositories, they do not support any notion of shared ownership. We consider this a severe limitation because contributing parties cannot jointly decide how their resources are used. In this paper, we introduced a novel concept of shared ownership and we described it through a formal access control model, called SOM. We then propose two possible instantiations of our proposed shared ownership model. Our first solution, called Commune, relies on secure file dispersal and collusion-resistant secret sharing to ensure that all

access grants in the cloud require the support of an agreed threshold of owners. As such, Commune can be used in existing agnostic clouds without modifications to the platforms. Our second solution, dubbed Comrade, leverages the block chain technology in order to reach consensus on access control decision. Unlike Commune, Comrade requires that the cloud is able to translate access control decisions that achieved consensus in the block chain into storage access control rules. Comrade, however, shows better performance than Commune.

References

1. Sharing in MULTICS. In Proceedings of the Fourth Symposium on Operating System Principles, SOSP 1973, Thomas J. Watson, Research Center, Yorktown Heights, New York, USA, October 15-17, 1973.

2. Robert Morris and Ken Thompson. Password Security: A Case History, 1979. http://cs-www.cs.yale.edu/homes/arvind/cs422/doc/unix-sec.pdf.

3. Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In Dan Boneh, editor, Advances in Cryptology – CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings, volume 2729 of Lecture Notes in Computer Science, pages 617–630. Springer, 2003.

4. Password Hashing Competition (PHC), 2014. https://password-hashing.net/index.html.

5. Donghoon Chang, Arpan Jati, Sweta Mishra, and Somitra Kumar Sanadhya. Rig: A simple, secure and flexible design for password hashing. In Dongdai Lin, Moti Yung, and Jianying Zhou, editors, Information Security and Cryptology - 10th International Conference, Inscrypt 2014, Beijing, China, December 13-15, 2014, Revised Selected Papers, volume 8957 of Lecture Notes in Computer Science, pages 361–381. Springer, 2014.

6. Ari Juels and Ronald L. Rivest. Honeywords: making passwordcracking detectable. In 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4- 8, 2013, 2013.

7. Fred Cohen. The Use of Deception Techniques: Honeypots and Decoys. http://all.net/journal/deception/Deception Techniques .pdf.

8. Lance Spitzner. Honeytokens: The Other Honeypot, 2003. http://www.symantec.com/connect/articles/honeytokens-other-honeypot.

9. Hristo Bojinov, Elie Bursztein, Xavier Boyen, and Dan Boneh. Kamouflage: Loss-resistant password management. In Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings, pages 286–302, 2010.

10. Wikipedia contributors. 2012 LinkedIn hack. Wikipedia, The Free Encyclopedia, Date retrieved: 29 May 2016. Available at: https://en.wikipedia.org/w/index.php?title=2012 LinkedIn

hack&oldid=722095159.
11. Bruce Schneier. Cryptographic Blunders Revealed by Adobe's Password Leak. Schneier on Security, 2013. Available at: https://www.schneier.com/blog/archives/2013/11/ cryptographic b.html.
12. Swati Khandelwal. Hacking any eBay Account in just 1 minute, 2014. Available at: http://thehackernews.com/2014/09/ hacking-ebay-accounts.html.
13. Wikipedia contributors. Ashley Madison data breach. Wikipedia, The Free Encyclopedia, Date retrieved: 29 May 2016. Available at: https://en.wikipedia.org/w/index.php?title= Ashley Madison data breach&oldid=721001290.
14. Troy Hunt. Observations and thoughts on the LinkedIn data breach, 2015. Available at: https://www.troyhunt.com/ observations-and-thoughts-on-the-linkedin-data-breach/.
15. Michael Gilleland. Levenshtein Distance, in Three Flavors. Available at: http://people.cs.pitt.edu/_kirk/cs1501/assignments/editdistance/Levenshtein%20Distance.htm.

15. Bibliography

(1) Java Complete Reference by Herbert Shield
(2) Database Programming with JDBC and Java by George Reese
(3) Java and XML By Brett McLaughlin
(4) Wikipedia, URL: http://www.wikipedia.org.
(5) Answers.com, Online Dictionary, Encyclopedia and much more, URL: http://www.answers.com
(6) Google, URL: http://www.google.co.in
(7)Project Management URL: http://www.startwright.com/project.html