International Journal of Multidisciplinary Research and Development

International Journal of Multidisciplinary Research and Development

Volume: 2, Issue: 8, 543-546
Aug 2015
www.allsubjectjournal.com
e-ISSN: 2349-4182
p-ISSN: 2349-5979
Impact Factor: 3.762

M. Sri vidya
Research Scholar,
Vivekanandha College of Arts and Sciences for Women (Autonomous), Namakkal, Tamilnadu, India.

S. Dhanalakshmi
Professor & Head,
Department of Computer Science & Applications,
Vivekanandha College of Arts and Sciences for Women (Autonomous), Namakkal, Tamilnadu India.

Correspondence
M. Sri vidya
Research Scholar,
Vivekanandha College of Arts and Sciences for Women (Autonomous), Namakkal, Tamilnadu, India.

Multi-Level Comparison of Data De-duplication with Message Encryption

M. Sri vidya, S. Dhanalakshmi

Abstract
Data access control is an effective way to ensure the data security in the cloud. Due to data outsourcing and un trusted cloud servers, the data access control becomes a challenging issue in cloud storage systems. Cipher text-Policy Attribute based Encryption (CP-ABE) is regarded as one of the most suitable technologies for data access control in cloud storage, because it gives data owners more direct control on access policies. However, it is difficult to directly apply existing CP-ABE schemes to data access control for cloud storage systems because of the attribute revocation problem.
In this thesis, we design an expressive, efficient and revocable data access control scheme for multi-authority cloud storage systems, where there are multiple authorities co-exist and each authority is able to issue attributes independently. Specifically, we propose a revocable multi-authority CP-ABE scheme, and apply it as the underlying techniques to design the data access control scheme. Our attribute revocation method can efficiently achieve both forward security and backward security. The analysis and simulation results show that our proposed data access control scheme is secure in the random oracle model and is more efficient than previous works.

Keywords: attribute revocation method, Cipher text-Policy Attribute based Encryption, multi-authority, forward security.

1. Introduction
DE-KEY operates in a cluster setting in which multiple hosts are directly connected to a single, shared SCSI volume and use a file system designed to permit symmetric and cooperative access to the data stored on the shared disk. DE-KEY itself runs on each host as a layer on top of the file system, taking advantage of file system block layout policies and native support for copy-on-write (COW) blocks. In this section, we provide a brief overview of our approach to deduplication and the file system support it depends on. DE-KEY uses content hashes to identify potential duplicates, the same basic premise shared by all deduplication systems. An index stored on the shared file system and designed for concurrent access permits efficient duplicate detection by tracking all known blocks in the file system by their content hashes.
In order to minimize impact on critical file system operations such as reading and writing to files, DE-KEY updates this index out of band, buffering updates and applying them in large, periodic batches. As part of this process, DE-KEY detects and eliminates duplicates introduced since the last index update. This can be done as an infrequent, low priority background task or even scheduled during times of low activity. Unlike approaches to deduplication such as content-addressable storage that integrate content indexes directly into the file system storage management, DE-KEY's index serves solely to identify duplicate blocks and plays no role in general file system operations. DE-KEY divides this index update process between hosts. Each host monitors its own changes to files in the cluster file system and stores summaries of recent modifications in on-disk write logs. These logs include content hashes computed in-band, as blocks are written to disk.
Each host periodically consumes the write logs of files it has (or can gain) exclusive access to and updates the shared index to reflect these recorded modifications. In the process, it discovers and reclaims any block whose content is identical to the content of some previously indexed block. Having each host participate in the index update process allows the hosts to divide and distribute the burden of deduplication, while sharing the index allows hosts to detect duplicates even if they are introduced by separate hosts.
Out-of-band index updates mean DE-KEY must be resilient to stale index entries that do not reflect the latest content of recently updated blocks. Indeed, this is essentially unavoidable in a decentralized setting because of communication delays alone. While this means DE-KEY

generally must verify block contents when updating the index, this resilience has an important implication: DE-KEY's correctness does not depend on its ability to monitor every write to the file system. This has important performance benefits.

First, updates to write logs do not have to be crash-consistent with updates to file contents, which both simplifies fault tolerance and allows hosts to buffer updates to write logs to minimize additional IO. Second, this allows users to trade off the CPU and memory overhead of write monitoring for peak file system performance on a per-file basis. For example, a user could simply disable deduplication for VMs that are performance-critical or unlikely to contain much duplicate data. Finally, this allows the write monitor to shed work if the system is overloaded. Because DE-KEY operates on a live file system, it specifically optimizes for unique blocks (blocks with no known duplicates). Unlike shared blocks, these blocks remain mutable after deduplication. The mutability of unique blocks combined with DE-KEY's resilience to stale index information means these blocks can be updated in place without the need to allocate space for a copy or to synchronously update the index. As a result, deduplication has no impact on the performance of writing to unique blocks, a highly desirable property because these are precisely he blocks that do not benefit from deduplication. Similar to some other deduplication work related to virtual disks, DE-KEY uses fixed-size blocks. Unlike stream-oriented workloads such as backup, where variable-sized chunks typically achieve better deduplication, our input data is expected to be block-structured because guest file systems (e.g., ext3, NTFS) typically divide the disk into fixed-size 4 KB or 8 KB blocks themselves. Consistent with this expectation, earlier work and test results we use a block size of 4 KB.

Existing System

File systems hosting virtual machines typically contain many duplicated blocks of data resulting in wasted storage space and increased storage array cache footprint. Deduplication addresses these problems by storing a single instance of each unique data block and sharing it between all original sources of that data. While deduplication is well understood for file systems with a centralized component, we investigate it in a decentralized cluster file system, specifically in the context of VM storage. However, as users we may want our files to be encrypted. We may not want the storage provider to see our data. Even if we did trust the provider, we may legitimately worry about errant employees or the risk of server compromise by an external adversary. When users themselves are corporations outsourcing their data storage, policy or government regulation may mandate encryption. Conventional encryption, however, makes deduplication impossible.

This new paradigm of data hosting and data access services introduces a great challenge to data access control. Because the cloud server cannot be fully trusted by data owners, they can no longer rely on servers to do access control. Ciphertext-Policy Attribute-based Encryption (CP-ABE) is regarded as one of the most suitable technologies for data access control in cloud storage systems, because it gives the data owner more direct control on access policies.

Drawbacks in Existing System

- Chase's multi-authority CP-ABE protocol allows the central authority to decrypt allthe cipher texts, since it holds the master key of the system.
- Chase's protocol does not support attribute revocation.

Proposed System

In this thesis, we first propose a revocable multi authority CP-ABE scheme, where an efficient and secure revocation method is proposed to solve the attribute revocation problem in the system. In CP-ABE scheme, there is an authority that is responsible for attribute management and key distribution.

Our attribute revocation method is efficient in the sense that it incurs less communication cost and computation cost, and is secure in the sense that it can achieve both backward security (The revoked user cannot decrypt any new cipher text that requires the revoked attribute to decrypt)and forward security (The newly joined user can also decrypt the previously published ciphertexts1, if it has sufficient. attributes).

Our scheme does not require the server to be fully trusted, because the key update is enforced by each attribute authority not the server. Even if the server is not semi trusted in some scenarios, our scheme can still guarantee the backward security. Then, we apply our proposed revocable multi-authority CP-ABE scheme as the underlying techniques to construct the expressive and secure data access control scheme for multi-authority cloud storage systems.

Benefits of Proposed System

- We modify the framework of the scheme and make it more practical to cloud storage systems, in which data owners are not involved in the key generation.
- We greatly improve the efficiency of the attribute revocation method.
- We also highly improve the expressiveness of our access control scheme, where we remove the limitation that each attribute can only appear at most once in a cipher text.

Module Description

User Module

Admin

In this module is used to help the server to view details and upload files with the security. Admin upload the data's to database. Also view the subscriber details and user details. Admin find the redistribute details. Also who send the data and receive the data's. Data owner store large amount of data to clouds and access data using secure key provided admin after encrypting data's. Encrypt the data using SECY. User store data after auditor, view and verifying data and also changed data. User again views data at that time admin provided the message to user only changes data.

Provider

In this module subscriber choose document and download the data's from service providers. Subscribers pay the amount to service provider. Service provider provides that data key to subscriber. So subscribers download the data using data key. A cloud computing service provider serves users' service requests by using a server system, which is constructed and maintained by an infrastructure vendor and rented by the service provider.

User

In this module, Users are having authentication and security to access the detail which is presented in the ontology system. Before accessing or searching the details user should have the account in that otherwise they should register first user can register their details like user name, password, email, mobile no, and then. We develop this module, where the cloud storage can be made secure.

Security Model

In multi-authority cloud storage systems, we make the following assumptions:

- The CA is fully trusted in the system. It will not collude with any user, but it should be prevented from decrypting any cipher texts by itself.
- Each AA is trusted but can be corrupted by the adversary.
- The server is curious but honest. It is curious about the content of the encrypted data or the received message, but will execute correctly the task assigned by each attribute authority.
- Each user is dishonest and may collude to obtain unauthorized access to data.

We now describe the security model for our revocable multi-authority CP-ABE systems by the following game between a challenger and an adversary. Similar to the identity-based encryption schemes, the security model allows the adversary to query for any secret keys and update keys that cannot be used to decrypt the challenge cipher text. We assume that the adversaries can corrupt authorities only statically similar to key queries are made adaptively. Let SA denotes the set of all the attribute authorities.

Multi-Authority

Multi-authority access control scheme in the framework of data access control scheme for multi-authority cloud storage systems contains the following phases:

- PHASE-I - SYSTEM INITIALIZATION
- PHASE-II - SECRET KEY GENERATION BY AAS
- PHASE III - DATA ENCRYPTION BY OWNERS
- PHASE IV - DATA DECRYPTION BY USERS

Attribute Revocation (CP-ABE)

We propose a new revocable multi-authority CP-ABE protocol based on the single-authority CP-ABE proposed to multi authority scenario and make it revocable. We apply the techniques in Chase's multi-authority CP-ABE protocol to tie together the secret keys generated by different authorities for the same user and prevent the collusion attack. Specifically, we separate the functionality of the authority into a global certificate authority (CA) and multiple attribute authorities (AAs).

The CA sets up the system and accepts the registration of users and AAs in the system.7 It assigns a global user identity uid to each user and a global authority identity aid to each attribute authority in the system. Because the uid is globally unique in the system, secret keys issued by different AAs for the same uid can be tied together for decryption. Also, because each AA is associated with an aid, every attribute is distinguishable even though some AAs may issue the same attribute.

Cipher text-Policy Attribute-Based Encryption (CP-ABE) is a promising technique that is designed for access control of encrypted data. There are two types of CP-ABE systems: single authority CP-ABE where all attributes are managed by a single authority, and multi-authority CP-ABE, where attributes are from different domains and managed by different authorities. Multi-authority CP-ABE is more appropriate for the access control of cloud storage systems, as users may hold attributes issued by multiple authorities and the data owners may share the data using access policy defined over attributes from different authorities. However, due to the attribute revocation problem, these multi-authority CP-ABE schemes cannot be directly applied to data access control for such multi-authority cloud storage systems.

Conclusion

In this thesis work, we have proposed De-key, an efficient and reliable convergent key management scheme for secure deduplication. De-key applies deduplication among convergent keys and distributes convergent key shares across multiple key servers, while preserving semantic security of convergent keys and confidentiality of outsourced data. We implement De-key using the Ramp secret sharing scheme and demonstrate that it incurs small encoding/decoding overhead compared to the network transmission overhead in the regular upload/download operations. We formalize a new cryptographic primitive, Message-Locked Encryption (MLE), where the key under which encryption and decryption are performed is itself derived from the message. MLE provides a way to achieve secure deduplication (space-efficient secure outsourced storage), a goal currently targeted by numerous cloud-storage providers.

We provide definitions both for privacy and for a form of integrity that we call tag consistency. Based on this foundation, we make both practical and theoretical contributions. On the practical side, we provide ROM security analyses of a natural family of MLE schemes that includes deployed schemes. On the theoretical side the challenge is standard model solutions, and we make connections with deterministic encryption, hash functions secure on correlated inputs and the sample-then-extract paradigm to deliver schemes under different assumptions and for different classes of message sources. Our work shows that MLE is a primitive of both practical and theoretical interest.

References

1. R. Howe and Y. Matsuoka, "Robotics for surgery," Annu. Rev. Biomed. Eng., vol. 1, pp. 211–40, 1999.
2. V. Pekar, T. R. McNutt, and M. R. Kaus, "Automated model-based organ delineation for radiotherapy planning in prostatic region," Int. J. Rad. Oncol., Biol., Phys., vol. 60, no. 3, pp. 973–980, 2004.
3. A. Shimizu, T. Kimoto, H. Kobatake, S. Nawano, and K. Shinozaki, "Automated pancreas segmentation from three-dimensional contrast enhanced computed tomography," Int. J. Comput. Assist. Radiol. Surg., vol. 5, pp. 85–98, 2010.
4. M. Bellare, S. Keelveedhi, and T. Ristenpart, ''Message-Locked Encryption and Secure Deduplication,'' in Proc. IACR Cryptology ePrint Archive, 2012, pp. 296-3122012:631.
5. G.R. Blakley and C. Meadows, ''Security of Ramp Schemes,'' in Proc. Adv. CRYPTO, vol. 196, Lecture Notes in Computer Science, G.R. Blakley and D. Chaum, Eds., 1985, pp. 242-268.
6. A.T. Clements, I. Ahmad, M. Vilayannur, and J. Li, ''Decentralized Deduplication in San Cluster File Systems,'' in Proc. USENIX ATC, 2009, p. 8.
7. A. Rahumed, H.C.H. Chen, Y. Tang, P.P.C. Lee, and J.C.S. Lui, ''A secure Cloud Backup System with Assured Deletion and Version Control,'' in Proc. 3rd Int'l Workshop Security Cloud Comput., 2011, pp. 160-167.
8. J.R. Douceur, A. Adya, W.J. Bolosky, D. Simon, and M. Theimer, ''Reclaiming Space from Duplicate Files in a Serverless Distributed File System,'' in Proc. ICDCS, 2002, pp. 617-624.
9. J. Gantz and D. Reinsel, the Digital Universe in 2020: Big Data, Bigger Digital Shadows, Biggest Growth in the Far East, Dec. 2012. [Online]. Available:

http://www.emc.com/collateral/analystreports/idc-the-digital-universe-in-2020.pdf.
10. R. Geambasu, T. Kohno, A. Levy, and H.M. Levy, ‘‘Vanish: Increasing Data Privacy with Self-Destructing Data,’’ in Proc. USENIX Security Symp, Aug. 2009, pp. 316-299.
11. S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, ‘‘Proofs of Ownership in Remote Storage Systems,’’ in Proc. ACM Conf. Comput. Commun. Security, Y. Chen, G. Danezis,and V. Shmatikov, Eds., 2011, pp. 491-500.
12. D. Harnik, B. Pinkas, and A. Shulman-Peleg, ‘‘Side Channels in Cloud Services: Deduplication in Cloud Storage,’’ IEEE Security Privacy, vol. 8, no. 6, pp. 40-47, Nov./Dec. 2010.
13. S. Kamara and K. Lauter, ‘‘Cryptographic Cloud Storage,’’ in Proc. Financial Cryptography: Workshop Real-Life Cryptograph. Protocols Standardization, 2010, pp. 136-149.
14. M. Li, ‘‘On the Confidentiality of Information Dispersal Algorithms and their Erasure Codes,’’ in Proc. CoRR, 2012, pp. 1-4abs/ 1206.4123.
15. D. Meister and A. Brinkmann, ‘‘Multi-Level Comparison of Data Deduplication in a Backup Scenario,’’ in Proc. SYSTOR, 2009, pp. 1-12.
16. D.T. Meyer and W.J. Bolosky, ‘‘A Study of Practical Deduplication,’’ in Proc. 9th USENIX Conf. FAST, 2011, pp. 1-13.
17. M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl, ‘‘Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space,’’ in Proc. USENIX Security, 2011, p. 5.
18. W.K. Ng, Y. Wen, and H. Zhu, ‘‘Private Data Deduplication Protocols in Cloud Storage,’’ in Proc. 27th Annu. ACM Symp. Appl. Comput., S. Ossowski and P. Lecca, Eds., 2012, pp. 441-446.
19. R.D. Pietro and A. Sorniotti, ‘‘Boosting Efficiency and Security in Proof of Ownership for Deduplication,’’ in Proc. ACM Symp. Inf., Comput. Commun. Security, H.Y. Youm and Y. Won, Eds., 2012, pp. 81-82.
20. J.S. Plank, S. Simmerman, and C.D. Schuman, ‘‘Jerasure: A ibrary in C/C++ Facilitating Erasure Coding for Storage ApplicationsVVersion 1.2,’’ University of Tennessee, Knoxville, TN, USA, Tech. Rep. CS-08-627, Aug. 2008.