Proof of Shared Data And Construct Secured Agreement And Protection

Pushpadevi K[1], Mahalakshmi K[2], Meera R³, Pavunraj D⁴
[1, 2, 3, 4] P.S.R. Rengasamy College of Engineering for Women, Sivakasi, India

Abstract- *Cloud storage platforms promise a convenient way for users to share files and engage in collaborations, they require all files to have a single owner who unilaterally makes access control decisions. Existing clouds are, thus, agnostic to shared ownership. This can be a significant limitation in much collaboration because one owner can, for example, delete files and revoke access without consulting the other collaborators. To tackle this problem, a novel notion called Proof of Shared oWnership (PoSW), and proposes a specific PoSW scheme to implement both secure ownership verification and data deduplication. In the PoSW scheme, we employ the convergent encryption algorithm to protect the content confidentiality of the shared file, introduce the secret file dispersal and secret sharing algorithm to implement the shared authorization and file ownership, and construct a novel interaction protocol between the file owners and the cloud server to verify the shared ownership and achieve deduplication to the shared file. Security analysis and performance evaluation show the security and efficiency of the proposed scheme.*

Keywords- Shared ownership, Proof of Shared oWnership, Convergent encryption algorithm, Deduplication.

I. INTRODUCTION

Cloud computing is recognized as an alternative to traditional information technology [1] due to its intrinsic resource-sharing and low-maintenance characteristics. In cloud computing, the cloud service providers (CSPs), such as Amazon, are able to deliver various services to cloud users with the help of powerful data centres. By migrating the local data management systems into cloud servers, users can enjoy high-quality services and save significant investments on their local infrastructures. One of the most fundamental services offered by cloud providers is data storage. Let us consider a practical data application. A company allows its staffs in the same group or department to store and share files in the cloud. By utilizing the cloud, the staffs can be completely released from the trouble PoSW local data storage and maintenance. However, it also poses a significant risk to the confidentiality of those stored files. Specifically, the cloud servers managed by cloud providers are not fully trusted by users while the data files stored in the cloud may be sensitive and Confidential such as business plans. To preserve data privacy, a basic solution is to encrypt data files, and then upload the encrypted data into the cloud [2]. Unfortunately, designing an efficient and secure data sharing scheme for groups in the cloud is not an easy task due to the following challenging issues. 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 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 repository in the cloud 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, if there is a sole owner, then he can abuse his rights by unilaterally making access control decisions. The community features a number of anecdotes where malicious users revoke access to shared files from other collaborators. This problem is further exacerbated by users increasingly storing most of their data in the cloud without keeping local copies, and accessing them through portable devices that have limited storage capacity. 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.The notion of shared ownership within a file access control model named PoSW, and uses it to define a novel access control problem of distributed enforcement of shared ownership in existing clouds. To existing a first solution, called Commune, which distributive enforces PoSW and can be deployed in an agnostic cloud platform. Commune ensures that (i) aowned by the customer and located on-premise (i.e., in the customers region of control). In particular, this means that access to customer data is under its control and is only granted to parties it trusts. In a public cloud the infrastructure is owned and managed by a cloud service provider and is located o_-premise (i.e., in the service provider's region of control). This means that customer data is outside its control and could potentially be granted to untrusted parties. Storage services based on public clouds such as Microsoft's Azure storage service and Amazon's S3 provide customers with scalable and dynamic storage. By moving their data to the cloud customers can avoid the costs of building and maintaining a private storage infrastructure, opting instead to pay a service provider as a function of its needs. For most customers, this provides several benefits including availability (i.e., being able to access data from anywhere) and reliability (i.e., not having to worry about backups) at a relatively low cost. While the benefits of using a public cloud infrastructure are clear, it introduces significant security and privacy risks. In fact, it seems that the biggest hurdle to the adoption of cloud storage (and cloud computing in general) is concern over the confidentiality and integrity of data. While, so far, consumers have been willing to trade privacy for the convenience of software services (e.g., for web-based email, calendars, pictures etc), this is not the case for enterprises and government.

II. LITERATURE SURVEY

Openness and Security in Cloud Computing Service
Author:CHANGYONG LIANG, HEMANT K. JAIN AND DONGXIAO GU.

Cloud computing companies need to achieve a balancebetween service openness and security control to promote the growth of the cloud service business. This paper is a theoretical discussion of the complete service openness condition and the complete security control condition based on Nash equilibrium from the perspective of investment, including investment in service and investment in security by the cloud computing company.The quantitative assessment methods of two conditions are designed based on the investments.

A quantitative analysis of the impact of security investments on security control and service openness is presented.Finally, the relationship between them is analyzed to help cloud computing providers decide the optimal strategy for coordinating investment in both service and security.

Cloud computing threats and provide security assessment
Author: H. MPH YU, K. WILLIAMS.

Cloud computing has faced scrutiny regarding security risks involved with allowing sensitive data to be controlled and handled by third-party, off-site vendors.Cloud computing is an emerging technological paradigm that offers on-demand, scalable, resources and IT-based solutions without the need to invest in new infrastructure or train new personnel. Cloud computing has faced scrutiny regarding security risks involved with allowing sensitive data to be controlled and handled by third-party, off-site vendors. Many businesses with interest in using cloud services do not have a process to assess cloud providers security. In this paper we categorize cloud computing threats. Here external threats, guest-to-guest threats and cloud-to-guest threats are avoided. Shared data are corrupted in this assessment.

Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects
Author: HAMED PIRSIVASH DEVA, RAMANAN CHARLESS, C. FOWLKES.

The Greedy algorithms allow one to embed pre-processing steps, such as non-max suppression, within the tracking algorithm. Furthermore, we give a near-optimal algorithm based on dynamic programming which runs in time linear in the number of objects and linear in the sequence length. The video sequence object is merged. Multi object tracking focus on the data association. The random variable is not acquiring. The image border is always large too. This enables our system to tolerate one provider’s failure by an added overhead cost of approximately 10% instead of 100% when full content replication is used.

A View of Cloud Computing
Author: M. ARMBRUST, A. FOX, R. GRIFFITH, A.D.JOSEPH, R. KATZ, A. KONWINSKI, G. LEE.

The goalof this this paper to reduce that confusion by clarifying terms, providing simple figures to quantify comparisons between of cloud and conventional Computing, and identifying the top technical and non-technical obstacles and opportunities of Cloud Computing.

Applications Software needs to both scales down rapidly as well as scale up, which is a new requirement. Such software also needs a pay-for-use licensing model to match needs of Cloud Computing. Infrastructure Software needs to be aware that it is no longer running on bare metal but on VMs.Moreover,billing needs to build in from the start. Hardware Systems should be designed at the scale of a container (at least a dozen racks), which will be is the minimum purchase size. Cost of operation will match performance and cost of purchase in importance, rewarding energy proportionality such as by putting idle portions of the memory, disk, and network.

Proofs of Retrievability: Theory and Implementation
Author: K. BOWERS, A. JUELS, AND A. OPREA

A proof of retrievability (POR) is a compact proof by a file system (prover) to a client (verifier) that a target file F is intact, in the sense that the client can fully recover it. As PORs incur lower communication complexity than transmission of F itself, they are an attractive building block for high-assurance remote storage systems. In this paper, we propose a theoretical framework for the design of PORs. Our framework improves the previously proposed POR constructions of Juels-Kaliski and Shacham-Waters, and also sheds light on the conceptual limitations of previous theoretical models for PORs. It supports a fully Byzantine adversarial model, carrying only the restriction—fundamental to all PORs—that the adversary's error rate be bounded when the client seeks to extract F. Our techniques support efficient protocols. We propose a new variant on the Juels-Kaliski protocol and describe a prototype implementation. We demonstrate practical encoding even for files F whose size exceeds that of client main memory. We have proposed in this paper a new framework for theoretical design of POR protocols that incorporates existing POR constructions, and enables design of new protocols with a wide range of parameter trade-offs. We showed how the protocols of Juels-Kaliski and Shacham-Waters can be simplified and improved using the new framework. We designed a new variant of the Juels-Kaliski scheme that achieves lower storage overhead, tolerates higher error rates, and can be proven secure in a stronger adversarial setting. Finally, we provided a Java implementation of the encoding algorithm of the new variant, in which files are processed and encoded incrementally, i.e., as they are read into main memory.

III. RELATED WORKS

To the best of our knowledge, this is the first work to (i) formulate and solve the problem of distributed enforcement of shared ownership policies. In the following we survey relevant related work in the areas of information dispersal, all-or-nothing transformations, and access control. Secret Sharing and Information Dispersal Secret sharing schemes [3] allow a dealer to distribute a secret among a number of shareholders, such that only authorized subsets of shareholders can reconstruct the secret. In threshold secret sharing schemes [4], [5], the dealer defines a threshold t and each set of shareholders of cardinality equal to or greater than t is authorized to reconstruct the secret. Secret sharing guarantees security (i.e., the secret cannot be recovered) against a non-authorized subset of shareholders; however, they incur a high computation/storage cost, which makes them impractical for sharing large files

Ramp schemes [6] constitute a trade-off between the security guarantees of secret sharing and the efficiency of information dispersal algorithms. A ramp scheme achieves higher "code rates" than secret sharing and features two thresholds t1, t2. At least t2 shares are required to reconstruct the secret and less than t1 shares provide no information about the secret; a number of shares between t1 and t2 leak "PoSW" information. All or Nothing Transformations All-or-nothing transformations (AONTs) were first introduced in [7] and later investigated in [8], [9]. The majority of AONTs leverage a secret key that is embedded in the output blocks. Once all output blocks are available, the key can be recovered and single blocks can be reverted.

Access Control Systems Current state-of-the-art access control systems—such as SecPAL [10], KeyNote and Delegation Logic—can in principle express t out of n policies. These languages, however, rely on the presence of a centralized PDP component to evaluate their policies. Furthermore, their PDPs cannot be deployed within a third-party cloud platform. As explained in Section II, these access control systems rely on an administrator to define and modify access control policies. In our setting, this means that a set of owners has to elect one enforcer who has unilateral powers over their files.

SYSTEM ARCHITECTURE

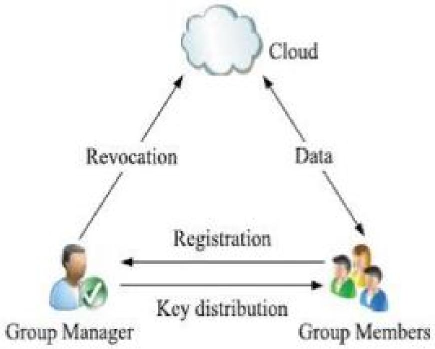

Fig. System Architecture

In the proposed system they considered the framework in presence of both node and link attack. They initially proposed an algorithm called JLNA which is used to reduce the β-disruptor problem. But they have used the sparse cut method which has more unwanted cuts. Hence, they propose an algorithm called Hole repair (HORA) algorithm.It is used control the difference between the connectivity in the residual graph and the target connectivity. In the proposed system we attain the reconstructed network to attain the maximum transmission without loss of the data and time consumption for the transmission. Here the process takes place with the high link cost and the less distance between each transmission nodes.

IV. MODULES

- Access Control Module
- Policy Module
- Document Module
- Collection Module
- Oblivious Cloud Ontology

ACCESS CONTROL MODULE

The Access Control Module has two sub-modules: one for Users and another for Groups. User Access Control limiting the access to user's information is also covered by Access Control modules. They can alter the list of existing users, hiding those who should not be discovered, or provide permissions to perform administrative actions not requiring full User Administration privileges.Group Access Control allows content to be organized into groups thatuserscan join. Access is controlled by group roles. The default roles are anonymous, outsider (logged-in but not member), and member, administrators can create additional group roles.

POLICY MODULE

The Policy Module is used to manage the consumer's policies on Encryption, Storage and Access Permissions. The Encryption policy module has all encryption policies defined for oblivious cloud store. Users can add/change encryption policy; it has all the fields defined by the ontology such as encryption algorithm, type, attribute/identity-based encryption etc. Encrypted fields in a database do nothing to protect information as it is accessed across the network.A policy module may view existing certificate properties and extensions, and it may also view request attributes and properties. Security is not just about the strength of encryption. Although this plays some role in any encryption policy, it needs to be considered as part of the whole. In some cases, ultimate strategies such as the hashing of data.

DOCUMENT MODULE

This module is used to manage the documents to be stored on the cloud obliviously. Document module shows list of documents that are stored in the system, collection to which the document belongs. On selecting a document from the list, it displays the details of that document such as collection, upload date, last modified date, user details, etc. When the details are entered by user, the save action initiates a process to store all the meta-data in the database and the actual document is stored in vORAM in HIRB. The access key when a new document is stored in oblivious cloud store. Note that, vORAM access key is shown for demo purpose only.

COLLECTION MODULE

The Collection form accepts a collection name, date of creation, a description of the collection, root key to access the collection that is being stored as HIRB instance. The policies defined for the collection are applied to its documents while storing them in vORAM. The vORAM stored all the collected data and their root keys as HIRB instance. The data are collected from user registration. The payment collection process involves pursuing payments of debts that have been owed by individuals cloud owners in the cloud service.

OBLIVIOUS CLOUD ONTOLOGY

In our ontology, every data object to be stored on the cloud is defined as a "document". The document can be of any format supported by the cloud provider. All security policies, such as encryption policy, oblivious storage policy etc. are applied at the Collection level. Controlling and limiting access to such documents requires a robust and trustworthy access control mechanism. The documents may be pdf, douc, audio, video, image etc. In this cloud ontology vORAM is used to store and process the cloud data.

V. SYSTEM ANALYSIS

A cloud computing architecture by combining with an example that a company uses a cloud to enable its staffs in the same group or department to share files. The system model consists of three different entities: the cloud, a group manager (i.e., the company manager), and a large number of group members (i.e., the staffs) as illustrated in Cloud is operated by CSPs and provides priced abundant storage services. However, the cloud is not fully trusted by users since the CSPs are very likely to be outside of the cloud users' trusted domain. Similar to, we assume that the cloud server is honest but curious. That is, the cloud server will not maliciously delete or modify user data due to the protection of data auditing schemes, but will try to learn the content of the stored data and the identities of cloud users. To fully ensure that the data integrity and save the cloud users computation resources as well as online burden, it is critical importance to enable public auditing service for cloud storage, which provides a much more easy and affordable way for the users to ensure their storage correctness in the cloud. The result from the TPAS would also be beneficial for the cloud service providers to improve their cloud-based service platform, even when serve for independent arbitration purpose. Distributed 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. The Shared-Ownership file access control Model (POSW) to define our

notion of shared ownership, and to formally state the given enforcement problem.

HORA(Hole Repair Algorithm)

HORA used to repair coverage holes of network, nodes with higher degree of density are moved to maintain uniform network density without increasing the coverage degree of the neighbor of a mobile node.The long-term cost of ownership may at first not seem to add up, but take into consideration other factors such as reduced risk and added value and for many organizations on-demand services make a lot of sense.

ALGORITHM:

```
DROP1(Training set T): Instance set S
Let S = T
For each instance P in S:
Find P.N..k+1, the k+1 nearest neighbors of P in S
        Add P to each of its neighbors' lists of associates
For each instance P in S:
        Let with = # of associates of P classified with P
        Let without = # of associates of Pclassified without P
        If (without - with) >= 0
                Remove P from S.
                For each associate A of P
                Remove P from A's list of nearest neighbors
        Find a new nearest neighbor for A.
                Add A to its new neighbor's list.
For each neighbor N of P.
                Remove P from its N's lists of associates.
Endif
 Return S.
```

AUDITING PROTOCOL

In cloud computing, data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Then, we extend our auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle model.The auditing protocol is extended to support the dynamic data operations, which is provable secure and efficient in the random oracle model. Also, the auditing protocol is extended to support batch auditing for multiple owners, without using any trusted organizer. The simulation and analysis results show that the proposed auditing protocols are efficient and secure, especially it reduces the computation cost of the auditor.Suppose a file F has m data components as F = (F_1,..., F_m). Each data component has its physical meanings and can be updated dynamically by the data owners.

KEY GENERATION ALGORITHM

Dynamic key is a unique encrypted key which is generated and sent to user during each login session. Client decrypts the key by using decryption application which uses the algorithm, there by client obtains the original key, this key is sent to the authentication server to get access to the cloud storage. It is an algorithm run by SM, which takes as input the public parameters params, master key and an identifier of either a user Ui ∈ U or the SP, and outputs a corresponding private key ski. This algorithm can be either probabilistic or deterministic.In this paper to implement a user authentication algorithm, which can be used in cloud storage to verify the authenticity of the user. In this paper we build a secure e-mail cloud-based algorithm, where the user's e-mail is used as an authentication, presenting a onetime encrypted password for the user and password is decrypted using proposed algorithm in user's e-mail application. In this paper we propose the user's e-mail as an authenticating device presenting a onetime password for the user.

VI. RESULT ANALYSIS

Based on the feedback received, there is massive support for the potential introduction of a Cloud style service by UIS. Aside from a couple of researchers who were fully invested in using Cloud Pro and didn’t believe there was any real benefit to them in adopting another solution, pretty much everyone else was very interested in the prospect (with many trying to push for further details of potential features, quotas and a launch date). The only other exceptions were a couple of Postdoctoral Research Associates who were also currently using Cloud Pro as a means of storing and backing up all of their data, and particularly liked the fact that they were not tied to the University in any way - and would still have full access to all of their data if/when they moved on from Cambridge at some point in the future. Although the potential for inclusion in a proof of concept / pilot was not a question raised during the discussions until fairly late in the project, everyone who was asked about this indicated they would be very happy to participate - which underlines the considerable appetite for such a solution.

VII. CONCLUSION

Even though existingcloud platforms are used as collaborative platforms, they surprisingly do not support any notion of shared ownership. We consider this to be a severe limitation because collaborators cannot jointly decide how their resources are used. The problem of enforcing shared ownership in the cloud is even more difficult since a cloud platform does not allow deployment of a third-party enforcement component.

VIII. ACKNOWLEDGMENT

I take his chance to precise my hearty due to my guide faculty member **PAVUNRAJ.D** for his steering and sharing his findings for technical steering and direction. Suggestions given by him were forever useful during this work to succeed. His leadership has been greatly valuable on behalf of me to figure on this project and is available with the best out of it.

REFERENCES

[1] JIAN XU 1,2, CHANGYONG LIANG1,3, HEMANT K. JAIN4, AND DONGXIAO GU 1," Openness and Security in Cloud Computing Services: Assessment Methods and Investment Strategies Analysis", Digital Object Identifier 10.1109/ACCESS.2019.2900889March 18, 2019.

[2] Y. Yao, J. Cao, S. Qian, and X. Wang, ``Resource scheduling for real-time analytical workflow services in the cloud," IEEE Access, vol. 6,pp. 5791057922, 2018.

[3] M. A. Lemaet al., ``Business case and technology analysis for 5g low latency applications," IEEE Access, vol. 5, pp. 59175935, 2017.

[4] T. Yu and M. Winslett, "A unified scheme for resource protection in automated trust negotiation," in Proc. of SP'03, 2003.

[5] A. Adya, W. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In OSDI, pages 1–14, December 2002.

[6] N. Alon, H. Kaplan, M. Krivelevich, D. Malkhi, and J. Stern. Scalable secure storage when half the system is faulty. In ICALP, 2000.

[7] T. Yu and M. Winslett, "A unified scheme for resource protection in automated trust negotiation," in Proc. of SP'03, 2003.

[8] C. Charnes, J. Pieprzyk, and R. Safavi-Naini. Conditionally secure secret sharing schemes with disenrollment capability. In ACM Conference on Computer and Communications Security (CCS), pages 89–95, 1994 A. Shamir. How to Share a Secret? In Communications of the ACM, pages 612–613, 1979.

[9] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in Proc. of CCS'05, 2005. R. L. Rivest. All-or-Nothing Encryption and the Package Transform. In International Workshop on Fast Software Encryption (FSE), pages 210–218, 1999.