ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

# Optimized Multi-Clouds using Shamir Shares

D.Mounica[#1], Mrs Ch.Radhika Rani[#2]

#1 Student, K.L.University,Vaddeswaram,Guntur(dt),

#2 Professor, K.L.University,Vaddeswaram,Guntur(dt),

**ABSTRACT:** Ensuring the security of cloud computing is a major factor in the cloud computing environment which has many benefits in terms of low cost and accessibility of data, as users often store sensitive information with cloud storage providers, unaware that these providers may be compromised. Dealing with “single cloud” providers is predicted to become less popular with customers due to risks of service availability failure and the possibility of malicious insiders in the single cloud. A movement towards “multi-clouds”, or in other words, “interclouds” or “cloud-of-clouds” has emerged recently and a system that employs Byzantine protocol for secret sharing has been constructed. We aim to equip Depsky framework to supply a secure cloud database that will guarantee to prevent security risks facing the cloud computing community. In relation to data intrusion and data integrity, like depsky we distribute the data and metadata into different cloud providers, and we apply the secret sharing algorithm on the stored data in the cloud provider. Instead of using plain secret sharing using public key ciphers we employ Shamir’s secret sharing algorithm. Hence, replicating data into multi-clouds by using a multi-share technique may reduce the risk of data intrusion and increase data integrity. This work aims to promote the use of multi-clouds due to its ability to reduce security risks that affect the cloud computing user.

## I INTRODUCTION

The use of cloud computing has increased rapidly in many organizations. Small and medium companies use cloud computing services for various reasons, including because these services provide fast access to their applications and reduce their infrastructure costs. Cloud providers should address privacy and security issues as a matter of high and urgent priority. Dealing with “single cloud” providers is becoming less popular with customers due to potential problems such as service availability failure and the possibility that there are malicious insiders in the single cloud. In recent years, there has been a move towards “multiclouds”, “intercloud” or “cloud-of-clouds” which is a solution to the malicious insider problem.

Later, DepSky System(Multi-Clouds Model) was proposed as a solution to the malicious insider problem. As data and information will be shared with a third party, cloud computing users want to avoid an untrusted cloud provider. Protecting private and important information, such as credit card details or a patient’s medical records from attackers or malicious insiders is of critical importance. Address three security factors that particularly affects single clouds, namely data integrity, data intrusion, and service availability using multiclouds. Technically DepSky systems contain

- Byzantine protocol - to integrate different clods
- Secret sharing - between different clouds, cloud provider and cloud user.
- Cryptography - for securing content.

Multi-clouds have the ability to decrease security risks that affect the cloud computing user.

In this paper, we aim to provide a framework to supply a secure cloud database that will guarantee to prevent security risks facing the cloud computing community. This framework will apply multi-clouds and the secret sharing algorithm to reduce the risk of data intrusion and the loss of service availability in the cloud and ensure data integrity. In relation to data intrusion and data integrity, like depsky we distribute the data and metadata into different cloud providers, and we apply the secret sharing algorithm on the stored data in the cloud provider. Instead of using plain secret sharing using public key ciphers we employ Shamir's secret sharing algorithm. An intruder needs to retrieve at least three values to be able to find out the real value that we want to hide from the intruder. This depends on Shamir's secret sharing algorithm with a polynomial function technique which claims that even with full knowledge of (k – 1) clouds, the service provider will not have any knowledge of vs (vs is the secret value). In other words, hackers need to retrieve all the information from the cloud providers to know the real value of the data in the cloud. Therefore, if the attacker hacked one cloud provider's password or even two cloud provider's passwords, they still need to hack the third cloud provider (in the case where k = 3) to know the secret which is the worst case scenario. Hence, replicating data into multi-clouds by using a multi-share technique may reduce the risk of data intrusion and increase data integrity.

## II RELATED WORK

Hore et al. [1] describes techniques for building privacy preserving indices on sensitive attributes of a relational table, and provides an efficient solution for data bucketization.

A scheme progressive elliptic curve encryption is presented in [2] that used multiple encryption keys to encrypt a part of data multiple time such that final cipher can be decrypted in one run using single key. This scheme is based on public/private key cryptography and consumers of apllication manage cryptographic private keys. Furthermore, N re-encryption will be required for N users in case of single data piece sharing.

Sion [3] emphasizes that a system that employs encryption for data outsourcing is secure if it ensures correctness, confidentiality and data access privacy. He also discusses about how these components are inter- related.

Williams and Sion [4], describe a new PIR technique developed using the Oblivious RAM (ORAM) technique. They illustrate the much lesser communication latency and computational complexity the technique exhibits, with a small increase in the space complexity.

A scheme on distributed key management is proposed in [5], which uses RSA algorithm for encryption/decryption. The main concept is to split key in multiple parts and divide among the group of users. If all users work on same text, a cipher text can be generated that will be equal to the cipher generated by actual key.

StrongAuth has StrongAuth KeyAppliance in [6] that uses third party library to develop an enterprise Key management Infrastructure, which support the services of public key infrastructure (PKI) as well as provides symmetric key management libraries. However, this library does not include any features that can securely manage keys at cloud platform. It requires a separate server for key storage and compromise of this server can create bottleneck for key security.

Re-Encryption based key management scheme is presented in [7] that is used in cloud based mobile application. Core concept of this scheme is the deployment of manager that can be an entity such as secure server between mobile device and cloud. Manager communicates to cloud as well as end users. It issue public, private key pairs for each user and maintain Access Control List (ACL) to enforce authorization. Public key repository for all users is maintained on cloud and any one from system user can access it but cannot decode it (as all private keys are maintaining by server). Users use their private key to encrypt any request and upload the cipher on cloud. Other users who require the data make a request to cloud controller. Cloud controller sends that request to manger. On the basis of Access Control List, manger decides to give the control of data to any user. First, it fetches the data from cloud storage and decrypts it using private key of the sender. Then manger will re-encrypt data with requester public key and send cipher text to requester. Requester decrypts the data with his/her private key and same process continues.

Hacigumus et al. [8] discusses a method for executing queries over encrypted data, at the service provider's site, and suggests splitting a query into two parts, namely the server query and client query. The server query is executed over the encrypted data at the service provider and the other part over the results of server query, at the client side.

## III SYSTEM MODEL AND FRAMEWORK

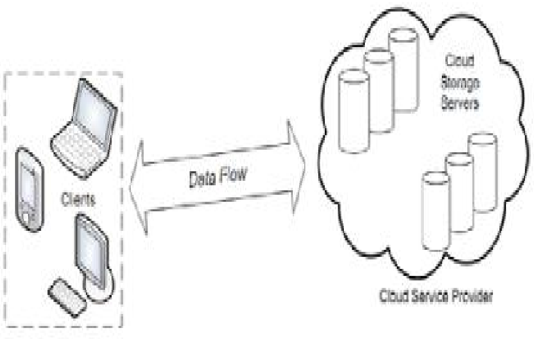

Fig-1. Cloud data storage architecture

The cloud data storage architecture used in this work is based on the model proposed by Cong Wang et. al, as shown in Fig - 1. The different entities are the Client and Cloud Storage Server.

Client: The end user who has large amount of data to store in the cloud and relies on the service provider for maintenance. This can either be an individual user or a large organization.

Cloud Storage Server: An entity, which is managed by a Cloud Service Provider, has significant storage space and computation resource to maintain client's data.

## IV SHAMIR'S SECRET SHARING ALGORITHMS

Data stored in the cloud can be compromised or lost. So, we have to come up with a way to secure those files. We can encrypt them before storing them in the cloud, which sorts out the disclosure aspects. Our goal is to divide some data D (e.g., the safe combination) into pieces D1, D2...,Dn in such a way that:

1. The Knowledge of any k or more Di pieces makes D easily computable.
2. The Knowledge of any k – 1 or fewer Di pieces leaves D completely undetermined (in the sense that all its possible values are equally likely).

This scheme is called (k,n)threshold scheme. If k=n then all participants are required to reconstruct the secret original data.

The essential idea of Adi Shamir's threshold scheme is that 2 points are sufficient to define a line, 3 points are sufficient to define a parabola, 4 points to define a cubic curve and so forth. That is, it takes k points to define a polynomial of degree k-1.

We divide our secret into pieces by picking a random degree polynomial $q(x)= a_0 + a_1 x+ a_2 x^2 +\ldots..+ a_{i-1} x^{k-1}$ in which $a_0= S$, $s_1= q(1)$, $s_2= q(2),\ldots\ldots, s_n = q(n)$ and represent each share as a point $(x_i , q(x_i) = y_i)$.

## V PROPOSED SYSTEM

Cloud customers may form their expectations based on their past experiences and organizations‟ needs. They are likely to conduct some sort of survey before choosing a cloud service provider. Customers are expected also to do security checks that are centered on three security concepts: confidentiality, integrity and availability.

Security in cloud services is based on the following:

- Strong network security is possible around the service delivery platform

- Data encryption: for data in transit (particularly over wide area networks), and sometimes stored data, but it cannot be applied to data in use.
- Access controls to ensure that only authorized users gain access to applications, data and the processing environment and is the primary means of securing cloud-based services.
- Service providers are able to inspect activity in their environment and provide reports to clients.

Logs need to be carefully constructed to appraisal the actions of their system administrators and other restricted users or risk producing reports that mix events relating to different customers of the service. In proposed system, replicating data into multi-clouds by using a multi-share technique [9] may reduce the risk of data intrusion and increase data integrity.

(a) DepSky System Model Architecture:

The DepSky system model contains three parts: readers, writers, and four cloud storage providers, where readers and writers are the client‟s tasks. Bessani et al. explain the difference between readers and writers for cloud storage. Readers can fail arbitrarily (for example, they can fail by crashing, they can fail from time to time and then display any behavior) whereas, writers only fail by crashing.

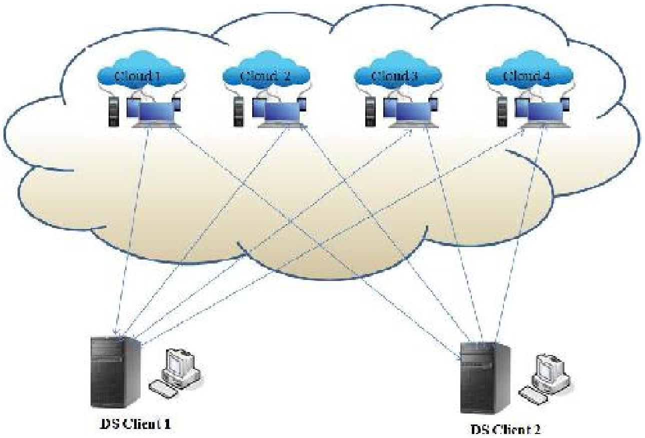

Fig-2: DepSky System Model Architecture

(b) Implementation:

- Data Integrity: It is not an easy task to securely maintain all essential data where it has the need in many applications for clients in cloud computing. To maintain our data in cloud computing, it may not be fully trustworthy because client doesn't have copy of all stored data. But any authors don't tell us data integrity through its user. So we have to establish proposed system for this using our data reading algorithm to check the integrity of data before and after the data insertion in cloud. Here the security of data before and after is checked by client with the help of CSP using our "effective automatic data reading algorithm from user as well as cloud level into the cloud" with truthfulness".
- Data Intrusion: The importance of data intrusion detection systems in a cloud computing environment. We find out how intrusion detection is performed on Software as a Service, Platform as a Service and Infrastructure as Service offerings, along with the available host, network and hypervisor - based intrusion detection options. Attacks on systems and data are a reality in the world we live in. Detecting and responding to those attacks has become the norm and is considered due diligence when it comes to security.
- Service Availability: Service availability is most important in the cloud computing security. Amazon already mentions in its licensing agreement that it is possible that the service might be unavailable from time to time. The user‟s web service may terminate for any reason at any time if any user‟s files break the cloud storage policy. In addition, if any damage occurs to any Amazon web service and the service fails, in this case there will be no charge to the Amazon Company for this failure. Companies seeking to protect services from such failure need measures such as backups or use of multiple providers

## VI CONCLUSION

In this paper, we aim to provide a framework to supply a secure cloud database that will guarantee to prevent security risks facing the cloud computing community. This framework will apply multi-clouds and the secret sharing algorithm to reduce the risk of data intrusion and the loss of service availability in the cloud and ensure data integrity. In relation to data intrusion and data integrity, we apply the secret sharing algorithm on the stored data in the cloud provider. Instead of using plain secret sharing using public key ciphers we employ Shamir's secret sharing algorithm. An intruder needs to retrieve at least three values to be able to find out the real value that we want to hide from the intruder. This depends on Shamir's secret sharing algorithm with a polynomial function technique which claims that even with full knowledge of (k – 1) clouds, the service provider will not have any knowledge of vs (vs is the secret value. Therefore, if the attacker hacked one cloud provider's password or even two cloud provider's passwords, they still need to hack the third cloud provider (in the case where k = 3) to know the secret which is the worst case scenario. Hence, replicating data into multi-clouds by using a multi-share technique may reduce the risk of data intrusion and increase data integrity.

## VII REFERENCES

[1] B. Hore, S. Mehrotra, and G. Tsudik, "A privacy - preserving index for range queries," in Proc. of the VLDB Conf., 2004, pp. 720 – 731.

[2] Gansen Zhao, Chunming Rongy, Jin Liz, Feng Zhangx and Yong Tang, "Trusted Data Sharing over Untrusted Cloud Storage Providers," 2nd IEEE International Conference on Cloud Computing Technology and Science.

[3] Sion, R.: Secure data outsourcing. In: Proc. of the VLDB Conf., pp. 1431– 1432 (2007).

[4] P. Williams and R. Sion, Usable PIR. NDSS, 2008.

[5] G. Zhao, S. Otenko, and D. Chadwick, "Distributed key Management for secure role based messaging," in Proceeding of The IEEE 20th International Conference on Advanced Information Networking and Applications (AINA2006), Vienna,Austria, April 2006.

[6] "An Introduction to Strong Key", white paper StrongAuth.Inc, October 2011.

[7] Piotr K. Tysowski, M.Anwarual Hasan, "Re-Encryption-Based Key Management towards Secure and Scalable Mobile Applications in Clouds".

[8] H. Hacigumus, B. R. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database service provider model," in Proc of the ACM SIGMOD Conf., 2002.

[9] M.A. AlZain and E. Pardede, "Using Multi Shares for Ensuring Privacy in Database-as-a-Service", 44th Hawaii Intl. Conf. on System Sciences (HICSS), 2011, pp. 1-9.