# SECURITY PROSPECTS THROUGH MULTICLOUD COMPUTING BY ADAPTING DATA SPLITTING

*SMITA Y.APARADH [1], MAHESH P.TAKALE [2]*

**[1] Ashokrao Mane Group of Institutions Wathar**
**[2] Sanjay Ghodawat Group of Institutions, Atigre**

***ABSTRACT:*** *Security is considered to be one of the most critical aspects in a cloud computing environment due to the sensitive and important information stored in the cloud for users. Users are wondering about attacks on the integrity and the availability of their data in the cloud from malicious insiders and outsiders, and from any collateral damage of cloud services. In cloud security and privacy can be achieved through the use of multiple distinct clouds simultaneously. Various distinct architectures are introduced and discussed according to their security and privacy capabilities and prospects. This paper provides a survey on the achievable security merits by adapting data splitting technique in multicloud architecture.*

***Keywords: Cloud, Multicloud, Security, Privacy, Data Spitting, Replication, Cryptography, Encryption***

## 1. INTRODUCTION

In cloud computing, the word cloud is used as a metaphor for ″the Internet,″ so the phrase cloud computing means "a type of Internet-based computing .Cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications. It relies on sharing computing resources.

Clouds can be categorized taking the physical location from the viewpoint of the user into account [2]. A public cloud is offered over the Internet and are owned and operated by a cloud provider. Public cloud services may be free or offered on a pay-per-usage model. Some examples include services aimed at the general public, such as online photo storage services, e-mail services, or social networking sites. However, services for enterprises can also be offered in a public cloud. In a private cloud, the cloud infrastructure is operated solely for a specific organization, and is managed by the organization or a third party. Some examples of privet computing are Eucalyptus, Elastra, VMware, and Microsoft. In a community cloud, the service is shared by several organizations and made available only to those groups. The infrastructure may be owned and operated by the organizations or by a cloud service provider. Such concerns might be related to regulatory compliance, such as audit requirements, or may be related to performance requirements, such as hosting applications that require a quick response time. A hybrid cloud is a combination of different methods of resource

The cloud computing service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Software as a service (or SaaS): is a way of delivering applications over the Internet—as a service. Instead of installing and maintaining software, you simply access it via the Internet, freeing yourself from complex software and hardware management. Platform as a Service (PaaS): Is a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider's infrastructure. To meet manageability and scalability requirements of the applications, It offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySQL and PHP), restricted J2EE, Ruby etc. Google's App Engine, Force.com, etc. are some of the popular PaaS examples. Infrastructure as a Service (Iaas): It provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data center space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, Go Grid, 3 Tera, etc. Basic Cloud Characteristics: The "no-need-to-know" in terms of the underlying details of infrastructure, applications interface with the infrastructure via the APIs. The "flexibility and elasticity" allows these systems to scale up and down at will utilizing the resources of all kinds CPU, storage, server capacity, load balancing, and databases. The "pay as much as used and needed" type of utility computing and the "always on! Anywhere and any place" type of network-based computing. Cloud are transparent to users and applications, they can be built in multiple ways branded products, proprietary open source, hardware or software, or just off-the-shelf PCs. In general, they are built on clusters of PC servers and off-the-shelf components plus Open Source software combined with in-house applications and/or system software.

## 2. CLOUD SECURITY ISSUES

Cloud computing creates a large number of security issues and challenges. A list of security threats to cloud computing is presented in [6]. There are numerous security issues for cloud computing as it encompasses many technologies including networks, databases, operating systems, virtualization, resource scheduling, transaction management, load balancing, concurrency control and memory management. Therefore, security issues for many of these systems and technologies are applicable to cloud computing. For example, the network that interconnects the systems in a cloud has to be secure. Furthermore, virtualization paradigm in cloud computing leads to several security concerns. For example, mapping the virtual machines to the physical machines has to be carried out securely. Data security involves encrypting the data as well as ensuring that appropriate policies are enforced for data sharing. In addition, resource allocation and memory management algorithms have to be secure. The main problem that the cloud computing paradigm implicitly contains is that of secure outsourcing of sensitive as well as business-critical data and processes. When considering using a cloud service, the user must be aware of the fact that all data given to the cloud provider leave the own control and protection sphere. Even more, if deploying data-processing applications to the cloud (via IaaS or PaaS), a cloud provider gains full control on these processes. Hence, a strong trust relationship between the cloud provider and the cloud user is considered a general. Prerequisite in cloud computing these cloud security issues and challenges triggered a lot of research activities, resulting in a quantity of proposals targeting the various cloud security threats. One promising concept makes use of multiple distinct clouds simultaneously.

## 3. SECURITY PROSPECT BY MULTICLOUD ARCHITECTURE

The basic idea is to use multiple distinct clouds to vanish or overcome the risks of malicious data manipulation, and disruptions in processes. By integrating many distinct clouds, the trust assumption can be lowered to an assumption of non-collaborating cloud service providers. By introducing multi
cloud it makes much harder Multicloud Architecture to enhance Security and Privacy for an external attacker to retrieve or damage the hosted data or applications of a particular cloud user. Many securities techniques and methods are adopted to solve the issues in the cloud. In multi cloud, cryptographic methods such as encryption and decryption and key management are used. Database splitting is one of the other important security techniques in involving a multi cloud Here we introduced four models

- Replication of applications
- Partition of application System into tiers
- Partition of application logic into fragments
- Partition of application data into fragments

### *3.1* Replication of application

It allows receiving multiple results from one operation performed in distinct clouds and to compare them within the own premise. This enables the user to get evidence on the integrity of the result. Instead of executing a particular application on one specific cloud, the same operation is executed by distinct clouds. By comparing the obtained results, the cloud user gets evidence on the integrity of the result. In such a setting, the required trust toward the cloud service provider can be lowered dramatically. Instead of trusting one cloud service provider totally, the cloud user only needs to rely on the assumption, which the cloud providers do not collaborate maliciously against it. (See Fig. 1)

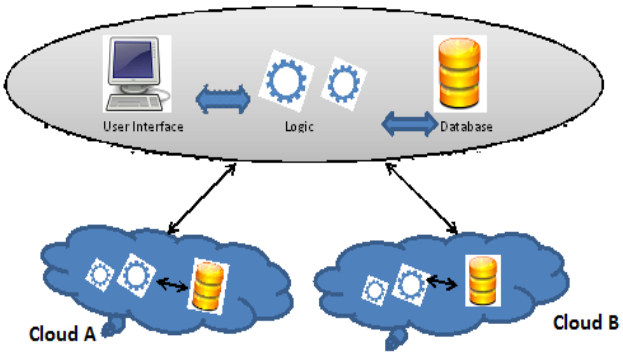


Fig. 1. Replication of application systems.

Assume that n > 1 clouds are available (like, e.g., Clouds A and B in Fig. 1). All of the n adopted clouds perform the same task. Assume further that f denotes the number of malicious clouds and that n _f > f the majority of the clouds are honest. The correct result can then be obtained by the cloud user by comparing the results and taking the majority as the correct one. There are other methods of deriving the correct result, for instance using the Turpin Coan algorithm [16] for solving the General Byzantine Agreement problem. Instead of having the cloud user performing the verification task, another viable approach consists in having one cloud monitoring the execution of the other clouds. For instance, Cloud A may announce intermediate results of its computations to an associated monitoring process running at Cloud B. This way, Cloud B can verify that Cloud A makes progress and sticks to the computation intended by the cloud user. As an extension of this approach, Cloud B may run a model checker service that verifies the execution path taken by Cloud A on-the-fly, allowing for immediate detection of irregularities. This approach might have a negative impact on the confidentiality because—due to the deployment of multiple clouds—the risk rises that one of them is malicious or compromised. To implement protection against an unauthorized access to data and logic this architecture needs to be combined with the architecture described in Section 3.2.

### *3.2* **Partition of application system into tiers**

It allows separating the logic from the data. This gives additional protection against data leakage due to the application logic. The architecture introduced in targets the risk of undesired data leakage. It answers the question on how a cloud user can be sure that the data access is implemented. To limit the risk of undesired data leakage due to application logic flaws, the separation of the application system's tiers and their delegation to distinct clouds is proposed (see Fig. 2). In case of an application failure, the data are not immediately at risk since it is physically separated and protected by an independent access control scheme. Moreover, the cloud user has the choice to select a particular—probably specially trusted—cloud provider for data storage services and a different cloud provider for applications.

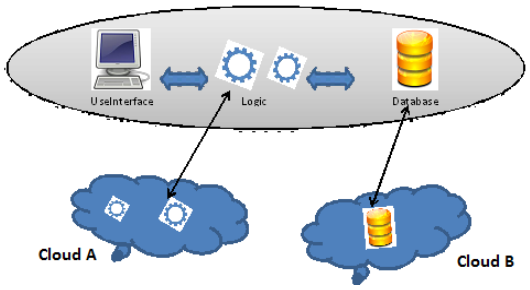


Fig.2 Partition of application system into tiers.

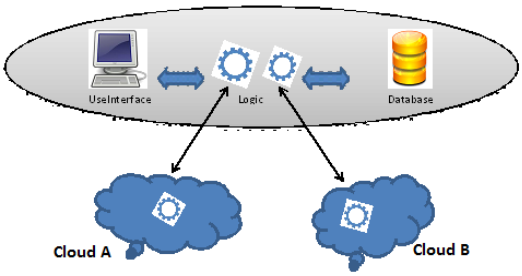


Fig. 3. Partition of application logic into fragments

### *3.3* **Partition of application logic into fragments**

This module allows distributing the application logic to distinct clouds. It has benefits. First, no cloud provider learns the complete application logic. Second, no cloud provider learns the overall calculated result of the application. Thus, this leads to data and application confidentiality. This architecture variant targets the confidentiality of data and processing logic. It gives an answer to the following question: how can a cloud user avoid fully revealing the data or processing logic to the cloud provider? The data should not only be protected while in the persistent storage, but in particular when it is processed [7]

### *3.4* **Partition of application data into fragments**

This multicloud architecture specifies that the application data is partitioned and distributed to distinct clouds (see Fig. 4). The most common forms of data storage are files and databases. Files typically contain unstructured data (e.g., pictures, text documents) and do not allow for easily splitting or exchanging parts of the data. This kind of data can only be partitioned using cryptographic methods (see Section 3.4.1).

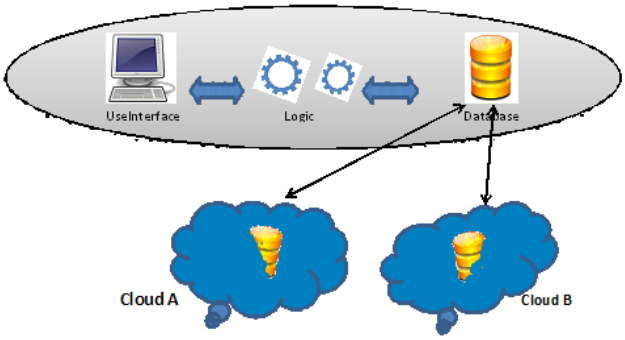


Fig. 4. Partition of application data into fragments.

Databases contain data in structured form organized in columns and rows. Here, data partitioning can be performed by distributing different parts of the database (tables, rows, columns) to different cloud providers (see Section 3.4.2). Finally, files can also contain structured data (e.g., XML data). Here, the data can be splitted using similar approaches like for databases. XML data, for example, can be partitioned on XML element level. However, such operations are very costly. Thus, this data are commonly rather treated using cryptographic data splitting.

**Data splitting by cryptographic method**

Probably, the most basic cryptographic method to store data securely is to store the data in encrypted form. While the cryptographic key could remain at the user's premises, to increase flexibility in cloud data processing or to enable multiuser systems it is beneficial to have the key available online when needed [6]. A similar approach is taken by several solutions for secure Cloud storage: The first approach to cryptographic cloud storage [8] is a solution for encrypted key/value storage in the cloud while maintaining the ability to easily access the data. It involves searchable encryption [9], [10] as the key component to achieve this. Searchable encryption allows keyword search on encrypted data if an authorized token for the keyword is provided. The keys are stored in a trusted private cloud whereas the data resides in the untrusted public cloud. Cryptographic data splitting has multiple advantages over current, widely used security approaches because:

- Enhanced security from moving shares of the data to different locations on one or more data depositories or storage devices (different logical, physical or geographical locations
- Shares of data can be split physically and under the control of different personnel reducing the possibility of compromising the data.
- A rigorous combination of the steps is used to secure data providing a comprehensive process of maintaining security of sensitive data.
- Data is encrypted with a secure key and split into one or more shares
- Lack of a single physical location towards which to focus an attack

**Database splitting**

For protecting information inside databases, one has to distinguish two security goals: confidentiality of data items or confidentiality of data item relationships. For splitting a database table, there are two general approaches: Vertical fragmentation and horizontal fragmentation [7]. With vertical fragmentation, the columns are distributed to cloud providers in such a way that no single provider learns a confidential relationship on his own. A patient health record, for example, might be fragmented into two parts, e.g., (name, patient number) and (patient number, disease). This way, the individual providers only learn noncritical data relations. However, for real-world applications, it is a nontrivial task to find such a fragmentation. First, new relations can be learned by performing transitive combination of existing ones. Second, some relations can be concluded using external knowledge. If, in the example above, the first provider additionally learns about the relation (patient number, medication), he has technically still no knowledge about the patient's disease. However, someone pharmaceutical background can derive the disease from the medication. Further, new relations can also be derived by combining multiple data sets. For instance, using again the relation of (patient number, medication), the knowledge of a combination of medications can ease the guessing of the patient's disease. Thus, also on a row level, database splitting might be required. This is called horizontal fragmentation. Finally, database splitting can also be combined with encryption. Using key management mechanisms like mentioned before, some database columns are encrypted. The combination of encryption and splitting protects confidential columns and still allows querying database entries using plain text columns. The benefits of a split database are **improved performance:** The performance of the database usually improves significantly because only the data

is sent across the network. In a shared database that is not split, the database objects themselves — tables, queries, forms, reports, macros and modules — are sent across the network, not just the data. **Greater availability**: Because only the data is sent across the network, database transactions such as record edits are completed more quickly, which leaves the data more available to edit. **Enhanced security:** if you store the back-end database on a computer that uses the NTFS file system, you can use NTFS security features to help protect your data. Because users access the back-end database by using linked tables, it is less likely that intruders can obtain unauthorized access to the data by stealing the front-end database or by posing as an authorized user. By default, Windows XP, Windows Vista, and Windows Server 2003 use the NTFS file system. If you are not sure what file system your file server uses, ask the system administrator. If you have administrator privileges on the file server, you can run the msinfo32 command to determine the file system yourself.

## 4. CONCLUSION

Cloud Computing is a relatively new concept that presents a good number of benefits for its users; however, it also raises some security problems which may slow down its use. The use of multiple cloud providers for gaining security and privacy benefits is nontrivial. As can be seen from the discussions of the four major multicloud approaches, each of them has its pitfalls and weak spots, either in terms of security guarantees, in terms of compliance to legal obligations, or in terms of feasibility. Given that every type of multicloud approach falls into one of these four categories, this implies a state of the art that is somewhat dissatisfying. In participation of application data into fragments, cryptographic data splitting and database splitting techniques are used. For splitting a database table, there are two general approaches: vertical fragmentation and horizontal fragmentation are used. The benefits of a split database are improved performance, Greater availability ,Enhanced security However, given their excellent properties in terms of security and compliance in multicloud architectures, we envision these fields to become the major building blocks for future generations of the multicloud computing paradigm.

## REFERENCES

[1] Jens-Matthias Bohli, Nils Gruschka, Meiko Jensen, Member, IEEE,Luigi Lo Iacono, and Ninja Marnau," Security and Privacy-Enhancing Multicloud Architectures" IEEE Transactions On Dependable And Secure Computing, Vol. 10, No. 4, July/August 2013

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing, Version 15," Nat'l Inst. of Standards and Technology, Information Technology Laboratory, vol. 53, p. 50, http://csrc.nist.gov/groups/SNS/cloud-computing/, 2010.

[3] D. Hubbard and M. Sutton, "Top Threats to Cloud Computing V1.0," Cloud Security Alliance, http://www Cloud security alliance.org/top threats, 2010

[4] R. Turpin and B.A. Coan, "Extending Binary Byzantine Agreement to Multivalued Byzantine Agreement," Information Processing Letters, vol. 18, no. 2, pp. 73-76, 1984.

[5] M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "On Technical Security Issues in Cloud Computing," Proc. IEEE Int'lConf. Cloud Computing (CLOUD-II), 2009.

[6] F. Pagano and D. Pagano, "Using In-Memory Encrypted Databases on the Cloud," Proc. First Int'l Workshop Securing Services on the Cloud (IWSSC), pp. 30-37, 2011.

[7] L.Wiese, "Horizontal Fragmentation for Data Outsourcing with Formula-Based Confidentiality Constraints," Proc. Fifth Int'l Workshop Security (IWSEC '10), pp. 101-116, 2010.

[8] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. 14th Int'l Conf. Financial Cryptography and Data Security, pp. 136-149, 2010.

[9] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," Proc. 13th ACM Conf. Computer and Comm. Security, pp. 79-88, 2006.

[10] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions," Proc. 25th Ann. Int'l Conf. Advances in Cryptology (CRYPTO '05), pp. 205-222, 2005.