On Securing Multi-Clouds: Survey on Advances and Current Challenges

Tara Salman

Abstract

Nowadays, storing and accessing data in multi-cloud infrastructure is being a common solution adopted by large organizations. Such adaptation would make organizations store their large amount of data in an efficient way without worrying about acceding their storage limits. It provides them with a flexible and dynamic storage that can grow and shrink based on the current need for data storage, and they pay based that. Besides, it provide them with the gain of multiple services from different clouds. Hence, it is been a cost-effective solution especially when organizational data storage needs change dynamically through different time of the year. However, that solution comes with some challenges such as the lack of management and security issues. Security is one of the biggest challenges that face such platforms and it is defined in terms of data privacy, availability, confidentiality and access control enforcements. In this paper, we discuss some of the recent advances to provide multi-cloud security and highlight their challenges and future research directions. This aims to understand the current trend in terms of complexity and strength of a secured solution and provide some insights of what is still left in such area of research.

Index Terms

Multi-clouds, single clouds, data privacy, data availability, cloud security, cloud architectures

I. Introduction

Since its starting in 2006, cloud computing have taking a lot of research interests and industrial implementations. The reason why this is being so popular goes back to organizational needs and adaptation for cloud services including storage, platforms and other services that are offers by cloud providers [1]. Small, medium and even large organizations are not buying their own storage and mitigating their data to the cloud as such solution is being an easier, scalable and more cost effective than hosting their own storage [2]. It is easy as organizations do not need to worry about their storage hardware and team anymore, scalable as they can scale and shrink their data storage without buying or changing hardwares and cost effective as they will not need to buy any hardware and they normally pay per storage units [3]. In addition, clouds can provide full functioning platforms to organizations allowing them to build their own specific platform and share it with others with worrying about their communication as soon as they are subscribed to the cloud [2]. Hence, cloud computing is being very popular and largely separated especially with the increase usage of internet connectively and virtualization techniques.

Recently, with the increase usage of clouds, organizations start adapting and subscribing to more than more cloud which introduces the concept of multi-cloud computing in both research and academic fields. Governmental and private organizations are putting lots of concern in this new innovation and its impact on their usage [4] [5]. A

simplest example of a multi-cloud system would happen if an organization need both public and private cloud. It can build its own private cloud and host it inside their organization or mitigate it to trusted cloud provider. At the same time, it adopt a public cloud for a large amount of data that does not need to be secured. Another example would happen if an organization have a large amount of data that can not be hosted on one small cloud provider so they mitigate to multiple cloud in order to support their needs. However, such strategies would come with a lot of challenges that includes guaranteeing organizational security aspects and the lack of management and control as the data is too separated.

One of most critical challenges in multi-clouds systems is security levels and how that impact the complexity of the system. Security in cloud computing comes in terms of data privacy, availability, confidentiality, integrity and access control mechanisms [6]. Data privacy include securing the data in rest and securing their access and their movement from clients to clouds or vise versa. Data availability is the ability to retrieve data by authorized users whenever possible. Integrity means that they should be retrieved correctly even if some of the data were corrupted. In addition to all that, accessing the data should be controlled and monitored by the organization and service providers so that no unauthorized used can access or alter private data [6]. Another challenging point that gets rigorous to control with multi-cloud issue is cloud trust as they are hosting important data. In such systems, clients need to trust multiple providers and multiple providers need to trust each others. Thus, data that are stored in the clouds can be accessed and retrieved securely with out lots of delay and without multiple cloud providers accessing the data of each others [7].

Existing solution to provide multi-cloud security can be loosely classified into: secured key based approaches [8], efficient distribution based approaches [9], and a hybrid solutions that combine both private and public data [10]. In this paper, we present multi-cloud security challenges and the current adopted solutions in achieving security with their advantages and disadvantages. This will open up a discussion of what can be done and added to multi-cloud and if the current implementations are efficiently secured and practically implementable.

The rest of the paper is organized as fallows: section II presents single cloud innovations, security requirements and solutions. Section III discuss multi-cloud in terms of the need for movement, architecture, security issues and thread introduced in such systems. Section IV would discuss possible, state of the arts solutions to achieve such security. Section V presents the weakness and strength of the current implementation and the left-over research problems to be solved in multi-cloud. At last, section VI would conclude our discussion and highlights expected future work in such area of research.

II. Single Clouds

In this section, we start by classifying single clouds in terms of multiple layers, including deployment, delivery and characteristics. Then, we go through single cloud security requirement and some famous breaching examples that happen over the last decade. This would motivate the work toward multi-cloud which will be explained in the next section.

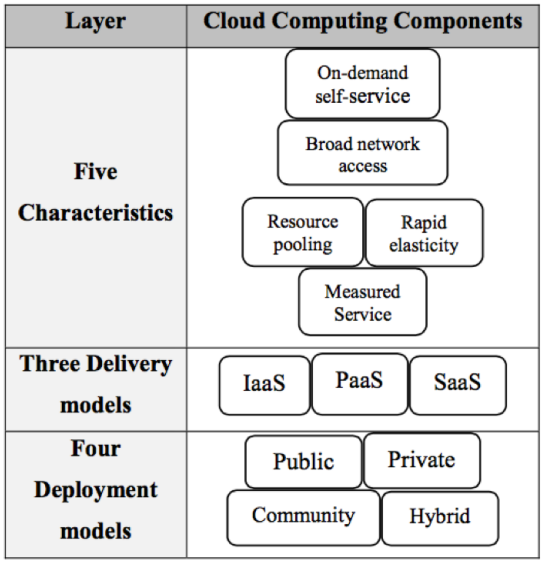


Fig. 1: Classification of Cloud in different layers [11]

A. *Cloud Classification*

Clouds can be classified in terms of their deployment model, their delivery model and their characteristics. Fig. 1 will summarize those classifications while the rest of this section would explain each one in brief.

In terms of deployment model, clouds can be classified into private, public, community and hybrid clouds [12]. A private cloud is available to only one user while secured from other accesses. A community cloud is a private cloud that is available to group of users. Public cloud are freely available and accessible by any device. Finally, a hybrid cloud is a combination of any two or more from public, private or community clouds.

Based on their deliveries, clouds can provide three models: data storage and computing facilities in Infrastructure as a service (IaaS) model, external runs of costumer applications in providers resource in Platform as a service (PaaS) model or provide services of licensed user's application in Services as a service (SaaS) model [12]. An example of Iaas is amazon EC2 services, while Paas is GoogleAPP service and SaaS is salesforce.com services.

Cloud characteristics that costumers are usually looking for are: On demand self service which give customers the resources as they need and charge them for that only, Continuous network access, Resource pooling at any time, elasticity of the storage and services they need and measurement services that indicate how many storage, cost, requirements each costumer is having [13].

B. *Cloud Security Requirements*

Security requirements in multi-cloud systems can be divided into:

- **Access control and its management:** Access management is set of rules that are defined by providers for their costumer access which guarantee their security. A provider, like a company or system owner, should have the ability to enforce their access policies and and manage them so that they can differentiate between

different user levels and privileges. Besides, providers should have some provisions of improper data access in cloud so that they recognize not well behaved costumers.

- **Data privacy:** providers need to guarantee their data privacy in which unauthorized user don't get access to confidential data.
- **Data integrity:** providers need to have some data integrity in which data can be retrieved and understood even if some portion of data is damaged.
- **Data intrusion:** Another security requirement by providers is data intrusion in which unauthorized access can not be made even if the passport is hacked.
- **Data availability:** Providers need to guarantee availability in which users can access data at any time within their provider rules and policies, i.e. data can not be accessed if the user violates any of the provider rules.

It is challenging to achieve these requirements in clouds as the data is stored and accessed through the internet which is anticipated to hacking no matter what security scheme is used. An attacker that has access confidentiality can alter and modify the data without costumers or even cloud providers knowledge [14]. In system like Paas, users are responsible for their data security which put them in a lot of risk as they are unaware of other costumers capabilities [13]. Beside that, a cloud provider, such as Amazon, might announced in their license agreement that the service might go down which effect data availability [15].

C. Single Clouds Famous Breaching Examples

Some examples of breaching schemes that occurred in last decade did happen to famous cloud providers such as Amazon and Google. In this subsection, we share some of the known attacks and point the reader to [16] for some more. For example:

- Signature wrapping attacks, [17], have proven their feasibility in [18] when implemented on EC2 frameworks. In such attacks, the eavesdropper, or communication listener, add a secondary random operation to the message while keeping the signature fixed. Such change will not be detected by EC2 framework and the operation will be executed on behalf of the victim account.
- In [19], the author attacked EC2 system by virtualization of the IaaS systems in which the attacker replace the physical machine by a virtualized one and use VM side channel attacks.
- [20] has reported an attack to Google Docs happened in 2009 which can be characterized as SaaS attacks example. Google Docs allow used to share documents and edit them with others by accessing online websites. Once a document is shared with anyone, it can be accessed by everyone who shared a document with the owner in previous. As such, unauthorized people can have the access to private data which defeat the privacy constrain to achieve security.
- Some other attacks presented in [21] [22] illustrated that major cloud provider have harsh security flaws in different cloud categories

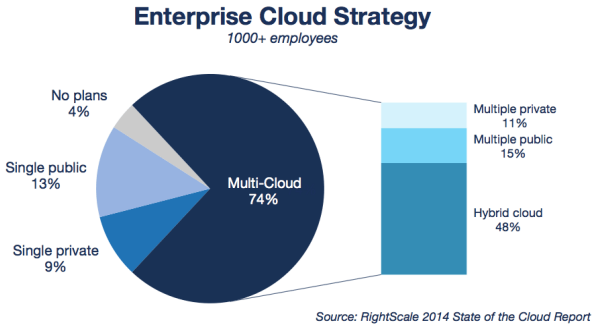


Fig. 2: Enterprise Usage of Single and Multi-Clouds [23]

III. Multi-Cloud

A multi-cloud system, also known as cloud-to-cloud or mashup clouds, is a distributed system where data have a certain degree of redundancy and replicated among different clouds owned by different vendors. An example of that is IBM mashup center, which a platform for sharing and reusing applications by web application tools. Appirio is a IaaS system that allow users to store their data in multiple Amazon C3 clouds using Salesforce.com clouds.

A. Movement to Multi-Cloud

Multi-Clouds have been widely used and adapted in both research and industrial fields over the past couple of year. To illustrate its current usage, Fig. 2 shows the usage of multi-cloud in compared to single cloud in 2014. As can be seen, multi-cloud is taking about 74% of enterprises usage which prove the large adaptation of such system in organizations [23].

This large movement to multi-clouds can be reasoned by the ability to separate private and public data, the dynamic data storage size that is needed, and the need for secondary services that are host on other clouds [24]. An organization might have both private and public data and in this case they will need multi-cloud to build a hybrid cloud, which constructed about 50% in Fig. 2. In this case, having a multi-cloud system can provide a private access in one cloud and a public access in another without mixing the two and allowing the IT team to concentrate more on securing their private data without worrying about their public ones.

Another reason to use multi-clouds is that stored data might have dynamic sizes based on different time of the year and hence need a dynamic infrastructure and a multi-cloud system would be the solutions for such scenarios.

A third reason might be the need for a service offered by another cloud in another part of the world. In this case, the costumer can either duplicate such service in his own cloud which would cost him money and consume time. An alternative can be subscribing to this service from the other cloud, in a multi-cloud manner, which would save the costumer a lot of time and might even cost less.

B. Multi-Cloud Architecture

A multi-cloud system can have multiple architectural views but they all share the same components: clients, cloud storage servers (CSS) and a manager. A client is the entity that is trying to store a large amount of data

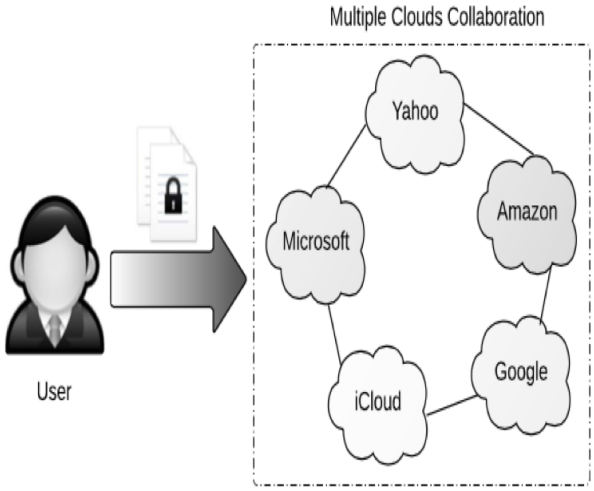


Fig. 3: Simple Multi-Cloud Architecture [26]

which can be either individuals or companies. A CSS is the entity that store the data and managed cloud service providers. A manager is the entity, or a broker, that maintain and enforce access control to data either by key or access control policies [25]. The manager can be located inside the organization or outsources to a trusted entity. A simplest architecture is shown in Fig. 3 where the manager is hosted inside the client and not shown in the figure.

In traditional architecture, a client that wishes to use multiple services from multiple cloud would manage his own access to both clouds and interact with each one individually, process the gathered data and generate his own results, as was shown in Fig. 3 . Due to tight cloud and client coupling, heterogeneity problems, pre-agreements with multiple clouds and the delivery model, this problem was inefficient enough to be solved by clients [4].

Hence, solutions have been proposes to automate the collaboration process by having a proxy between clients and providers that takeoff clients responsibilities in dealing with multiple clouds. A proxy can be owned, controlled and hosted by clients, can be a service that is offered by another cloud, can be within the organizational structure, managed by other entities or organization, or a hybrid solution that merge two or more approaches depending on client needs [4]. A sample architecture of such solution is shown in Fig. 4, where the broker is outsourced to a trusted server, data is divided into parts, stored in Cloud A and B, and the matadata is stored privately in a private cloud.

It should be noted that IT and cloud providers do offer both type of solutions where the client control its data separation and uploading to the cloud, as in Fig. 3, or an outsourced entity do have that control, as in Fig. 4. It is up to the enterprise to choose what is best for its application in this case. For example, Cisco gives the two solution which are Enterprise manage, or manage by cloud customer, and Service provider managed, which is managed by the cloud provider [27].

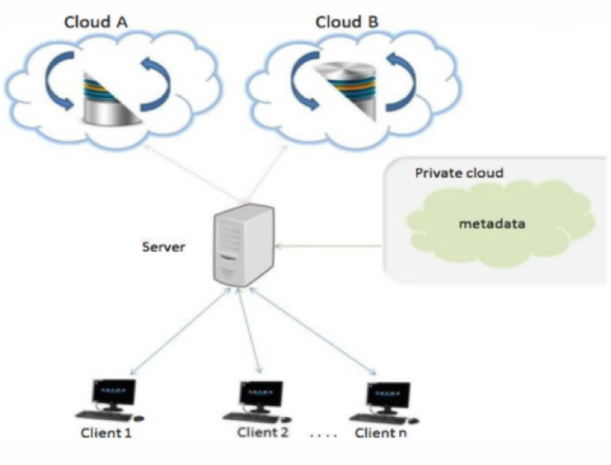


Fig. 4: Sample Solution for Traditional Multi-Cloud Architecture [8]

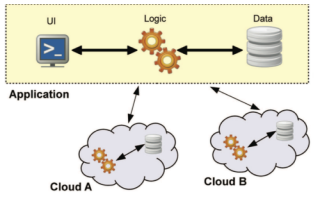
C. Multi-Cloud as a Secured Architecture

As a multi-cloud architecture, it can offer some level of security by allowing one or more of the fallowing four strategies, as explained in [6] and visualized in Fig. 5:

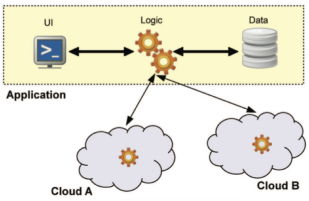
- **Data replication:** Data is duplicated in multiple clouds, as shown in Fig. 5a, and synchronized whenever a change happen. Replication allows receiving multiple copies of the same data and hence a better guarantee for data integrity and availability.
- **Data partitioning:** Partitioning is splitting the data between multiple clouds, as shown in Fig. 5b, such that no cloud provider can get a meaningful insights of the data that it is hosting. Such strategy would guarantee data privacy to some extent and does not allow data accessing by anyone other than authorized clients.
- **Application partitioning into tiers:** Partitioning application system into tiers, as shown in Fig. 5c, would allow the separation between application logic and data which gives additional data leakage protection due to application logic flow.
- **Application Fragmentation:** Fragmenting the application, as shown in Fig. 5d, would allow it to be saved on multiple clouds which provides a distributed fine grained fragments among multiple clouds. Thus, cloud provider can not gain access to full application, which safeguard data confidentiality.

D. Security Threads in Multi-Clouds

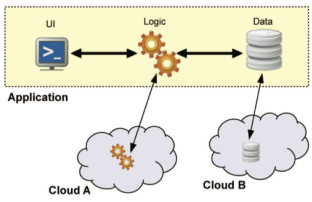
Even though multi-clouds systems provide some level of security, they comes with cost of configuration complexity and threads that were not introduced before. Configuration complexity comes with establishing trust between multiple entities and cloud providers and maintain data privacy while providing information to brokers and multiple other clouds. Threads include increasing attack possibility due to complexity of managements, loss of client controls which is moved to brokers, exposed interference due to public domain data storage and data privacy due to multi-tenancy [4][28].



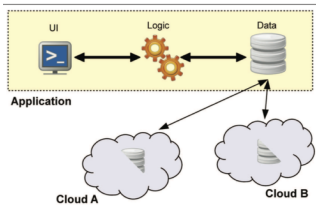
(a) Data Replication



(b) Data Partitioning



(c) Application Partitioning into tiers



(d) Application Fregementation

Fig. 5: Four Security Strategies offered By Multi-Clouds [6]

Additionally, the traditional security requirements still exist which include: securing stored data, in motion data, availability, and privacy, discussed in section II-B. However, the additional challenge over here is to manage that on multiple clouds and the security tradeoff with efficiency and fast responses when collecting from different clouds.

IV. Existing Solutions

A. *Theoretical solution*

In [4], the authors present three problems in multi-cloud security and propose some theoretical solutions for them. The first problem is the trust problem, or how can trust be build between multiple entities, which is one of the most important challenges in multi-clouds. They proposed to build a trust based on-fly-agreements and expected behavior, secure delegation to proxies by validating proxies using signatures or simple public keys algorithms.

The second problem is policy conflicts between multiple clouds and how can proxies assure meeting the requirement for each cloud. This would mean that multiple cloud could have different policy toward one user or a piece of data, which would result in a conflict. Proxies need to solve such conflicts and assure that each cloud requirements is met and the authorized user get his data whenever needed. They propose to use adaptive algorithms to solve such conflicts, build a flexible and scalable conflict resolution mechanisms, in situation of multiple conflicts use correlation mechanisms to identify dependent relationships and solve the problem among conflicting segments of resources.

The third and last presented problem was data privacy and how client can keep their privacy while providing enough information to proxies. They propose to use adaptive data perturbation which provides the tradeoff between privacy and utility of query results. Such methods should return highly accurate results for large number of records however with high noise to few important records, that guarantees data privacy.

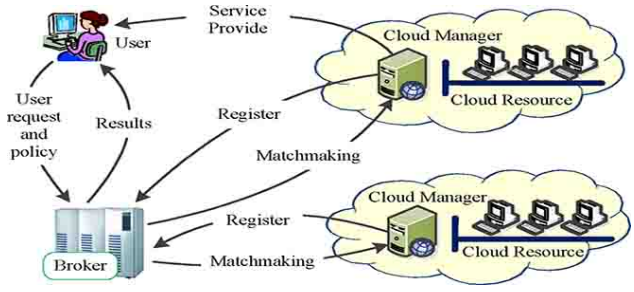


Fig. 6: SOTS System Architecture [30]

These solutions might be good in theory, however, implementation-wise, it is unclear how signature or public key algorithms, adaptive conflict solving algorithms or adaptive perturbation algorithm can be generated in such distributed scenario.

B. Distribution Based

1) Secure Cost-Effective Multi-Cloud Storage: As proposed in [9], secure cost-effective multi-cloud storage technique is based on minimizing a cost function based on linear programming [29]. The cost function is subject to a given maximized quality of services (QS) achieved at the time of retrieval. Having, p cloud providers, each with a cost C per data unit and QoS level for storage services, and N chunks of data divided by the users, where at least k number of chunks is needed to understand the data. At the same time q out of p clouds should take part in information retrievals to provide security. Then the minimization problem can be formulated as:

$$minimize[w_1 \sum_{i=1}^{N}(r_i * QS_i) - w_2 \sum_{i=1}^{p}(s_i * Ci)] \tag{1}$$

$$where \quad w_1 + w_2 = 1$$

$$subject \quad to \sum_{i=1}^{N} r_i = q \quad , \quad \sum_{i=1}^{p} s_i = N$$

$$and \quad q = k \leq N \leq p$$

This can provide a certain level of privacy as cloud can not have access to full data and to understand the data a hacker need to access k chunks from multiple clouds. However, the security can be breached easily by accessing k chunks. Hence k, N, q and p should be very carefully designed such that it imposes a rigorous hardness to understand the data without knowing their exact location, which is assuringly known to trusties only.

2) Service Operator-aware Trust Scheme (SOTS): In this scheme, as presented in [30], a system, called SOTS, for resource matching in multiple clouds is built. In such system, a trusted broker relies between client and cloud providers. The broker evaluate each resource periodically and add new resources that just joined the network.

As shown is Fig. 6, the system consist of four mean modules that the broker should do which are: adaptive trust evaluation modules, resource matching and distribution module, agent publish and resource acquisition model and resource register module. The adaptive trust module dynamically sort resources based on their high performance

and their historic high trust information. The resource matching model negotiate with new resource mangers for service level agreement (SLA) and rules to be assigned to their resources. The agent publish model guarantee that SLA is being fallowed and monitor resources allocation in the cloud. The resource register module manages the resources indexing in multiple clouds.

To get a resource, the client should send a query to the broker which will evaluate and match resources and return back the required resource. The broker should check for authentication, authorization and self-security competence (SEC) for each resource and client involved in the process

In this way, the resources are matched to multiple clouds in an efficient way and accessing any resource would require authentication and authorization which guarantee a certain level of security. However, the problem of such system is having trusted broker that is trusted by all clients, resource providers and multiple clouds.

C. Cryptography Based

1) Storage and Retrieval (STRE) Algorithm: An algorithm, as presented in [26], that allow users to store their encrypted data among multiple clouds while keeping their reliability guarantees. They provide an efficient secret sharing mechanism that allow better protection of data search patterns compared to existing techniques. Besides, they have a low overhead compared to baseline encryption algorithm but with a cost of communication overhead.

The protocol consist of three phases: setup, storage and retrieval. In the setup phase, the cryptography keys in initialized on the assumption that there exist a public key infrastructure. In the storage phase, the data is divided into chucks of equal sizes, encrypted and distributed on multiple clouds, each chunk of a cloud. In this stage, data is stored with redundancy in order to guarantee the later reliability. When retrieving data, the user initialize the query and the system encrypts them and send to n clouds. The user can get back the query results if t clouds $(t < n)$ reply with the results such that it is enough for the user to understand and decrypt the data. When unauthorized users tried to get the data, they can share the queries, however, the results back from clouds are not enough to understand the data as the encrypted data were not understood by the cloud providers.

To illustrate the process, Fig. 7 shows an example of how the algorithm will store and retrieve data. Assuming that the client have chunks A and B to store, C and D will be generated for redundancy. Then encrypted A, B,C, and D will be stored on different clouds. On the retrieving process, the client will send queries to all hosting clouds and some would reply with the encrypted data. Assuming that the client got back A and C or A and D while B was corrupted, he can safely retrieve B back by doing simple operations.

This can guarantee data privacy, availability and integrity assuming that half of the data can be retrieved. However, it comes with the cost of additional storage to store all the redundant chunks. Since the cloud provider charge organizations on their amount of data stored, such algorithms cost would at least doubled to handled all the redundant chunks. Hence, it would be cost inefficient especially if the organization has a large amount of data to store.

2) Identity-Based Distributed Provable Data Possession (ID-DPDP): As presented in [31], ID-DPDP is a protocol that provide a secure and efficient integrity check based on data encryption. The protocol consists of four algorithms: Setup, Extract, TagGen and Proof. In the Setup phase, the private and public master keys are generated with the

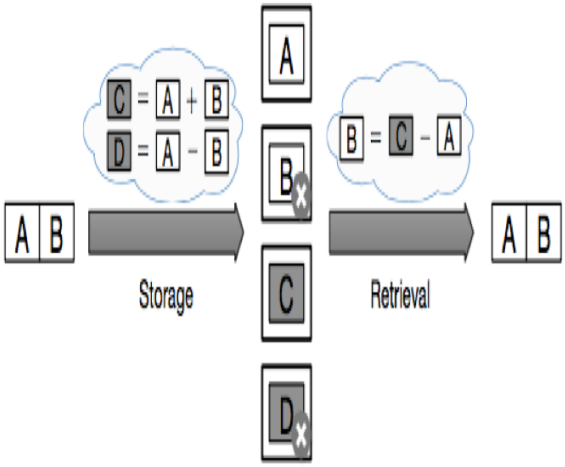


Fig. 7: STRE System Example [26]

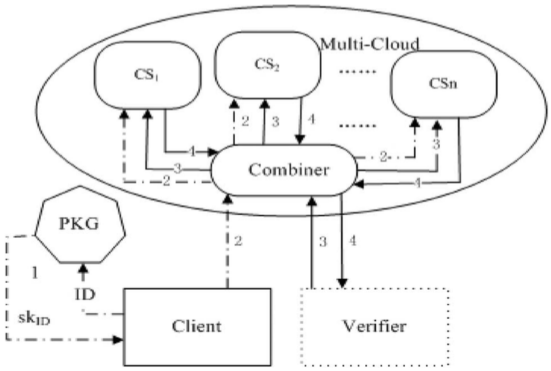


Fig. 8: ID-DPDP System Architecture [31]

public parameters indicated. The Extract phase would receive the master key, the public parameters and the identity of the client and extract a private key for this client with the given identity. The tagGen phase split the big chunk of data into multiple blocks, store them in different clouds, generate a tag for each block and return back the block-tag pairs. The Proof algorithm would take a challenge and its answer and return back if the challenge passed successfully or no.

To access a data, as shown in Fig. 8, the client gets it private key from Extract algorithm, create the block-tag pair from the TagGen algorithm and upload it to the Combiner whose the controller in this case. This reaches the verifier which sends challenges to the combiner that distribute the challenge among multiple clouds. Those clouds would reply back with their answers which is aggregated to the verifier again to ensure that the client have the access right.

Results showed that the algorithm has a limited computation power, can be implemented on mobiles, flexible and scalable due to its low computation and storage overhead. In terms of security, it can be achieved by verifying

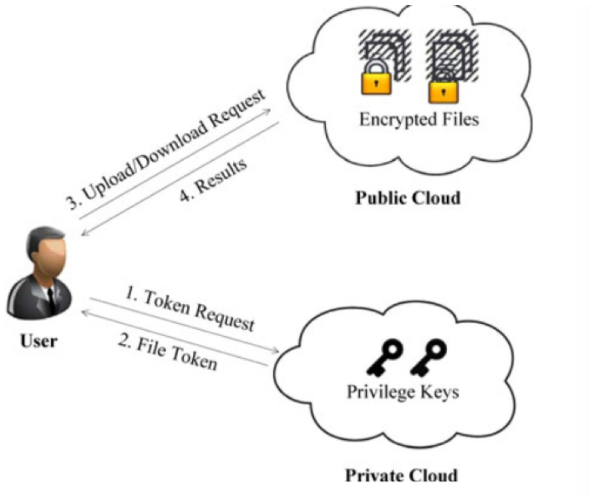


Fig. 9: Secure Authorized Deduplication Architecture [10]

the information each time it is queried and fail the request if it failed the verification. Moreover, it shows a low communication overhead which results in less complexity and reduce the security risks resulting from eavesdropping communications. However, it could achieve only integrity and privacy of data and the process of how the combiner can be trusted or located is still not discusses

D. Hybrid Based

1) A Hybrid Cloud Approach for Secure Authorized Deduplication: This scheme, as proposed in [10], explores the use of both private and public cloud to solve the problem of duplicates with different privileges access. The private cloud is used to store the keys for the files with specific privileges. In order to access a file, the user will need to have the key of that file and he need to be in a specific privilege. Hence, the private cloud is acting as an interface between the user and the public cloud. Fig. 9 highlights the protocol procedure in a simplified way while the rest of this subsection would explain it in details.

Uploading a file passes through a procedure which can be highlighted in the fallowing:

- First, when a client wants to store the data, he interacts with the private cloud to prove his identification with his privilege and private key. If the identification is passed, private cloud should find the corresponding privileges in its stored table.
- The user calculates and sends the file tag to the private cloud which return back the token to allow it to communicate with the public cloud. store its data in public cloud.
- The user consults the public cloud if his file is found in the public cloud, a duplicate, or it is a new file.
- If the file is new, the public cloud sends a signature to the user which is passed to the private cloud. The signature is first verified by the private cloud and sent to the other cloud with the same privileges. Furthermore, the user encrypts the file that he wish to upload and only then uploads it to the public cloud.

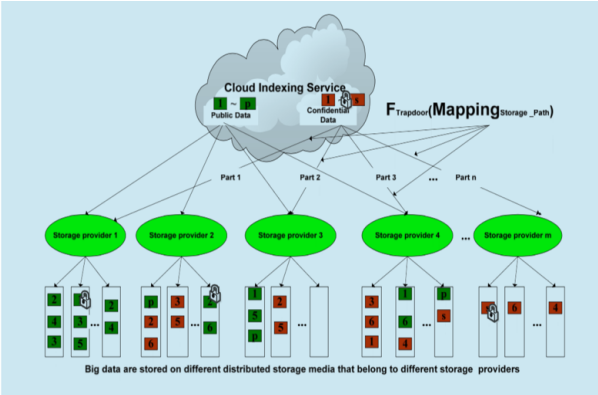


Fig. 10: Scure Big Data Storage Architecture [32]

- If the duplicate was actually found in public clouds, the user will need to verify with the public cloud that he has the write access, which is done by a challenge proof. If the proof is passed the user will get the pointer to the file with the signature and timestamp. The signature and the timestamp is uploaded to the private cloud which is first verified. If the verification is passed, the tag of the file is computed and uploaded to all public clouds with the privilege sets defined by the owner.

In order to retrieve a file, the user sends the request to the public cloud which checks for the privilege and if the user has the read access. If it passes the check, the public cloud will send the data encrypted and the user can decrypt it using his own private key.

In such scheme, the user can assure that their data is stored securely and no unauthorized privilege can get access to data, which guarantees data privacy up to some level. Even if something went wrong with the public clouds in the retrieving process, the user won't be able to decrypt the data as it was encrypted with another privilege key. However, such scheme comes with complex communication overhead during the storing process which adds delays and require key computation capabilities from clients, private clouds and public clouds.

2) Secure Big Data Storage and Sharing Scheme for Cloud Tenants: The authors in [32] provide a secure protocol to store big data that have both confidential and public privileges. The scheme protects the mapping of the confidential data into multiple clouds instead of protecting the data themselves. That is, as illustrated in Fig. 10, when receiving some big data, they divide it into multiple parts and store them in multiple clouds. The location of those chunks will be held encrypted in the cloud indexing service such that no one other than authorized users can access them. When accessing a data, the client should send the request with the data encryption key and the cloud will collect the data from different part and send it back to the user. If the key was not matched then the cloud indexing service can not access the confidential data as the location will be encrypted.

In this way, the data privacy can be guaranteed however data integrity and availability were not tackled. Besides, accessing the key, which is valid hacking strategy in some cases, can destroy the whole security purposes especially that all the location were encrypted with the same key.

V. Discussion

In this section, we summarize some discussion points on the pros and cons in the discussed algorithms. It should be noted that algorithms specific pros and cons are discussed within each algorithm discussion, however, some of the shared and obvious weaknesses and strength points are discussed in more details here.

First, achieving security is a tradeoff between how secure you need your data to be and how complex your algorithm should be. For example, we discussed the two hybrid solutions in section IV-D, in which both algorithms have the same objective, data encryption. The first scheme is too complex due to the storing communication overhead while the second scheme is relatively simple. One the other hand, the first one is more secured and can handle additional feature in duplication and access privileges control mechanisms. In cryptography based algorithm, presented in section IV-C, the second algorithm was definitely more cost and complexity efficient that the first algorithm. However, the first was simpler in terms of trust as it did not introduce the broker, or the combiner, which is an additional cost in terms to trust and needed to be trusted by everyone. The same thing is applied to distribution based approaches, presented in section IV-B, where the second algorithm provides a more secured solution but comes with the cost of broker trust.

In addition, hybrid based solutions did consider both private and public data which made it more realistic in industrail scenarios, however, hosting a private cloud was the additional cost that was not introduced in the others. Distribution based algorithms were simple as they did not involve any key generation or sharing which is a challenging task, especially in such distributed scenarios. STRE algorithm, in the cryptography based approaches, seems to be the easiest secured one to implement but its cost might be the highest due to the replication features.

Therefore, security in multi-cloud should be a tradeoff between complexity, level of security, cost, and other available resources. It is application and organizational specific to decide on such tradeoff based on the features it needs. Another tradeoff to think off is security requirement in terms of data privacy, integrity and availability.

A. Expected Future Directions

Most of the presented algorithms handle data privacy, integrity and availability issue with minimal work done on data intrusion and access control mechanisms. It is right that some did handle the access control in terms of the secret key management, however, if the key was broken, all the security will be breached. Thus, work done on data intrusion and access control mechanism is still inadequate and expected to be considered in a large scale in future research. In addition, Paas and Saas security have not been considered in the literature of multi-cloud system and expected to be tackled in future research.

Moreover, the trust between brokers, multi-clouds and clients is still an open area that have not been discussed yet. In such scenarios, the clients need to trust brokers, if exist, in distributing their data and handling their privacies. The broker can be either internally located, which is more secured but costly, or outsources outside the organization. The unsolved issue is how would the client trust brokers and to what level can the broker understand clients data. Besides, such architecture involve multiple cloud providers, accessed either by the clients or the broker, which might be unaware of each others. The question of to what extend should those providers trust each others and be aware of each has not been resolved yet.

Finally, the design of a complete solution that provide security requirements while keeping the cost and complexity in mind have not been clearly addressed and probably will be address again in future research in a more efficient ways.

VI. Conclusion

To summarize, multi-cloud systems have been widely used over the last half decade in both industry and research work. The reason why such systems are being the cloud trend goes back to the need of multiple clouds to support big data, multiple services and some level of security guarantees. However, such distributed architecture comes with some security challenges and threads which motivated the research work for multi-cloud security. This paper presented some recent advances and schemes to provide multi-cloud security which were lossy classified to: distributed, cryptography and hybrid based approaches. Distributed based approaches seems to be simple but provide less security that others. Hybrid based approaches were more realistically meeting organizational needs however they comes with private cloud costs. To empathize more, security can be defined as a tradeoff between complexity and security requirements and it is up to the users to decide what are best for them. The paper highlighted algorithms advantages and disadvantages beside some future expected work to be done in such area of research.

References

[1] T. Erl, R. Puttini, and Z. Mahmood, *Cloud Computing: Concepts, Technology, & Architecture*, ser. The Prentice Hall service technology series from Thomas Erl. Prentice Hall, 2013. [Online]. Available: https://books.google.com/books?id=zqhpAgAAQBAJ

[2] V. Chang, *Delivery and Adoption of Cloud Computing Services in Contemporary Organizations*, ser. Advances in Systems Analysis, Software Engineering, and High Performance Computing:. IGI Global, 2015. [Online]. Available: https://books.google.com/books?id=AWfCCAAAQBAJ

[3] K. Miller, "Enabling scalable and cost-effective ediscovery for customers ? zapproved, an apn technology partner," https://aws.amazon.com/blogs/apn/enabling-scalable-and-cost-effective-e-discovery-for-customers-zapproved-an-apn-technology-partner/, May 2015.

[4] M. Singhal, S. Chandrasekhar, T. Ge, R. Sandhu, R. Krishnan, G.-J. Ahn, and E. Bertino, "Collaboration in multicloud computing environments: Framework and security issues," *Computer*, vol. 46, no. 2, pp. 76–84, Feb 2013.

[5] H. Fard, R. Prodan, and T. Fahringer, "A truthful dynamic workflow scheduling mechanism for commercial multicloud environments," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 6, pp. 1203–1212, June 2013.

[6] J.-M. Bohli, N. Gruschka, M. Jensen, L. Iacono, and N. Marnau, "Security and privacy-enhancing multicloud architectures," *Dependable and Secure Computing, IEEE Transactions on*, vol. 10, no. 4, pp. 212–224, July 2013.

[7] H. Sato, A. Kanai, and S. Tanimoto, "A cloud trust model in a security aware cloud," in *Applications and the Internet (SAINT), 2010 10th IEEE/IPSJ International Symposium on*, July 2010, pp. 121–124.

[8] V. Balasaraswathi and S. Manikandan, "Enhanced security for multi-cloud storage using cryptographic data splitting with dynamic approach," in *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, May 2014, pp. 1190–1194.

[9] Y. Singh, F. Kandah, and W. Zhang, "A secured cost-effective multi-cloud storage in cloud computing," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, April 2011, pp. 619–624.

[10] J. Li, Y. K. Li, X. Chen, P. Lee, and W. Lou, "A hybrid cloud approach for secure authorized deduplication," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 5, pp. 1206–1216, May 2015.

[11] M. AlZain, E. Pardede, B. Soh, and J. Thom, "Cloud computing security: From single to multi-clouds," in *System Science (HICSS), 2012 45th Hawaii International Conference on*, Jan 2012, pp. 5490–5499.

[12] A. Bento, *Cloud Computing Service and Deployment Models: Layers and Management: Layers and Management*, ser. Premier reference source. Business Science Reference, 2012. [Online]. Available: https://books.google.com/books?id=IZ_V4uK49RMC

[13] H. Takabi, J. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," *Security Privacy, IEEE*, vol. 8, no. 6, pp. 24–31, Nov 2010.

[14] S. L. Garfinkel, "An evaluation of amazon's grid computing services: Ec2, s3 and sqs," Tech. Rep.

[15] Amazon, "Aws customer agreement," https://aws.amazon.com/agreement/, June 2015.

[16] T. Armding, "The 15 worst data security breaches of the 21st century — cso online," http://www.csoonline.com/article/2130877/data-protection/data-protection-the-15-worst-data-security-breaches-of-the-21st-century.html, Feb 2015.

[17] M. McIntosh and P. Austel, "Xml signature element wrapping attacks and countermeasures," in *Proceedings of the 2005 Workshop on Secure Web Services*, ser. SWS '05. New York, NY, USA: ACM, 2005, pp. 20–27. [Online]. Available: http://doi.acm.org/10.1145/1103022.1103026

[18] N. Gruschka and L. Iacono, "Vulnerable cloud: Soap message security validation revisited," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, July 2009, pp. 625–631.

[19] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-vm side channels and their use to extract private keys," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 305–316. [Online]. Available: http://doi.acm.org/10.1145/2382196.2382230

[20] J. Kincaid, "Google privacy blunder shares your docs without permission — techcrunch," http://techcrunch.com/2009/03/07/huge-google-privacy-blunder-shares-your-docs-without-permission/, March 2009.

[21] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "All your clouds are belong to us: Security analysis of cloud management interfaces," in *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, ser. CCSW '11. New York, NY, USA: ACM, 2011, pp. 3–14. [Online]. Available: http://doi.acm.org/10.1145/2046660.2046664

[22] S. Bugiel, S. Nürnberger, T. Pöppelmann, A.-R. Sadeghi, and T. Schneider, "Amazonia: When elasticity snaps back," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 389–400. [Online]. Available: http://doi.acm.org/10.1145/2046707.2046753

[23] K. Weins, "Cloud computing trends: 2014 state of the cloud survey," http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2014-state-cloud-survey, April 2014.

[24] M. Louk and H. Lim, "Homomorphic encryption in mobile multi cloud computing," in *Information Networking (ICOIN), 2015 International Conference on*, Jan 2015, pp. 493–497.

[25] S. Vishnupriya, P. Saranya, and A. Rajasri, "Secure multicloud storage with policy based access control and cooperative provable data possession," in *Information Communication and Embedded Systems (ICICES), 2014 International Conference on*, Feb 2014, pp. 1–6.

[26] J. Li, D. Lin, A. Squicciarini, J. Li, and C. Jia, "Towards privacy-preserving storage and retrieval in multiple clouds," *Cloud Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[27] "Cisco intercloud fabric: Hybrid cloud with choice, consistency, control and compliance," White Paper, Cisco, 2015.

[28] W. Jansen and T. Grance, "Sp 800-144. guidelines on security and privacy in public cloud computing," Gaithersburg, MD, United States, Tech. Rep., 2011.

[29] J. Strayer, *Linear Programming and Its Applications*, ser. Undergraduate Texts in Mathematics. Springer New York, 2012. [Online]. Available: https://books.google.com/books?id=XcTcBwAAQBAJ

[30] X. Li, H. Ma, F. Zhou, and X. Gui, "Service operator-aware trust scheme for resource matchmaking across multiple clouds," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 5, pp. 1419–1429, May 2015.

[31] H. Wang, "Identity-based distributed provable data possession in multicloud storage," *Services Computing, IEEE Transactions on*, vol. 8, no. 2, pp. 328–340, March 2015.

[32] C. Hongbing, R. Chunming, H. Kai, W. Weihong, and L. Yanyan, "Secure big data storage and sharing scheme for cloud tenants," *Communications, China*, vol. 12, no. 6, pp. 106–115, June 2015.