Cloud Computing Security and Solutions: Survey[*]

Emmanuel Akoja[†]
Louisiana State University
Department of Computer Science
Class:Advance Storage System
eakoja1@lsu.edu

ABSTRACT

Cloud computing is now the center of attention in storage systems over the past decades because of a massive expansion in demand for it. There are benefits to adopting cloud-based data storage systems. The advantages of cloud computing include basic technology, remote access, and the lower cost of management. However, the security and privacy challenges in cloud storage systems require critical attention. Scholars and industrial individuals have provided potential solutions to these challenges in the previous literature. This narrative review provides cloud security issues and requirements, identified threats, and compares strategic solutions presented by three published papers. This work aims to analyze the different solutions to security and confidentiality in cloud computing systems. Moreover, this work proposed a new idea on security solutions to improve data confidentiality. Additionally, this survey discussed several security challenges on cloud computing services and offered a potential solution. This paper will present detailed information about cloud security challenges such as cloud service providers, data owners, and cloud users. Keywords: Security · Threats · Vulnerabilities · Data protection

1. INTRODUCTION

Cloud computing is expanding because of its high-performance reports. Therefore, many big companies and individual users are moving their complex computing services to cloud servers to save space, time, cost, and storage. Outsourcing data to cloud servers is a solution to store extensive data into efficient distributed storage. Providers of cloud services have the infrastructure to host the data for their clients, and the clients can create, store, update, and access databases in real-time. A security challenge in outsourcing data to cloud servers is that the company loses its management to the provider. Five significant data aspects could be vulnerable;

Confidentiality: This describes the protection of data against malicious attack either internally or external. This is usually done by encrypting the data.
Integrity: This describes the protection against operations such as update, delete, insert by unauthorized users. Data integrity comprise of correctness (validation of result exists), completeness (no omission from results), and freshness (result based on last version)
Availability: This describe the protection against denial of service to users
Access control: Only authorized user should be able to access the outsourced data, and
Firewall: This is to protect the false accusation of claims by authorised users.

Five models can ease the above security concerns[5]:
Separation model: this separates the processing of data at different providers.
Availability model: This provides two providers for each data storage and ensures consistency at various storage providers.
Migration model: This creates a data migration service to move data from one storage provider to another.
Tunnel model: This creates a separate layer that separates data processing service from data storage service.
Cryptography model: This model encrypts the data before it is sent to storage providers.

These sets of models use duplication data, and separation of duty has proven to reduce integrity and availability challenges [13]. However, with encryption supervised by a provider, there is still the need to trust all data with the provider. Hence, the provider would have access to every cloud service and data.

This survey discusses the techniques developed to handle this set of vulnerabilities. If the provider monitors the storage and processing, the customer must trust the provider to all limits; this might not be comfortable with the client. This review summarises the method presented in the literature that reduces dependency on the provider. Some organizations like ENISA (European Network and Information Security Agency), CSA (Cloud Security Alliance), and NIST (National Institute of Standards Technology) have identified problems and vulnerabilities and provided possible solutions. The agencies mentioned above discussed security concerns but emphasized the classifications of services such as services in the form of Software (SaaS), Infrastructure as a service because the deployment models can easily apply to these services[2].

Another critical challenge is data misappropriation by cloud

[*] (A survey review for Advance Storage System class)

[†] Ph.D. Student

CSC7700 Advance Storage System

providers. Vital information such as the location of the virtual machine may be intruded on by an attacker. If providers guarantee the protection of clients' data, we can argue that clients can not be sure that providers keep their words.

The main goal of this review is to gather information from three papers to identify a list of security concerns in cloud computing, summarizes the solution provided to some of these concerns, identify some loopholes and weaknesses of these solutions, and give some opinions to improve the solution. The benefit of this review is to have a piece of organized, helpful information on this concern. The contribution of this work is to highlight the requirement of cloud computing, provide a comprehensive summary of algorithms for solutions to resolve problems, and proposing an idea that may cater to security concerns[10].

2. PREVIOUS WORKS

A scholar suggested that the introduction of Trust Platform Modules (TPM) at all datacenter can help to achieve security; clients would not be able to verify whether a storage site has TPM [3]. Dingledine et al. [1] described a distributed storage system of peer-to-peer data sharing called the Free Haven project. Free Haven does not assume an honest user and does not provide an avenue for an unknown user. The design was able to identify a dishonest user. Another distributed design is Oceanstore, but it is not protective against an illegal node[9]. Gonzalez et al. [2] identified some main problems and categorized them into seven models with subdivisions. The work provides a clear perspective of cloud computing problems. [2], these are:

Network security: This described the security challenge about network and configurations. A good measure to provide security to the network is to adopt local implementation of internal cloud services. Resource sharing results in more data transfer over the virtual machine; hence, VPN techniques are needed to protect the data and system from spoofing, side-channel attack, and sniffing. Creating a firewall will enable the isolation of virtual machines, prevent service denial, and detect external security assessment. And a secured configuration of technologies and protocols will create data system privacy with improved efficiency.

Interfaces security: This controls the cloud storage for both users and administrators. The use of API to access virtual resources, an administrative interface to remotely control services and resources, a user interfaces to explore the available resources with an integrated secured environment. An authentication set up to control access to the cloud.

Data security: Data security can guarantee availability, confidentiality, and integrity, as mentioned in the earlier section. There are three main requirements; Disposal techniques, including the complete deletion of data or having hidden backup registries. To avoid data loss and ensure availability, redundancy techniques could be a measure. Sensitive data can be secured using cryptography.

Virtualization issue: All hardware shares the same resources. If an attacker intrude one of the virtual machine, it may cause total leak of data. Hence, isolation could be a measure to avoid such attack. Some storage system uses hypervisor which can be prone to security bride. hence, there is need to provide a strong security at this level. Misuse of virtual machines can cause data leakage, wrong identification of a machine can also result in wrong execution of process.

Governance issue: Users gave all control to the cloud provider on a specific issue that may affect security. The administrator may lose power at some time due to security mechanisms or policies; the use of policies to restrict clients to some sensitive protocols and formats can prevent vulnerability of service termination. Providers' policies may not fully cover some situations.

In the paper written by Gonzalez et al.[2], the authors identify two main categories of works in the literature; the first is security framework, this king of work focused on compiling security and privacy risks and best approaches to avoid them. The second type of work is that which identify future trends and solutions.

3. MODELS AND ALGORITHMS

This section will be describing some models and algorithms found in the selected literature.

3.1 Botnets

Botnets comprise computers that monitor human botmaster using a convoluted hierarchy of computers to detect and disclose networks and their users[5]. Jatun et al. have extended this concept to split data and keep in several independent (non-colluding) storage in a redundant array of independent net-storages(RAIN). In this case, data are stored in several storage providers, and a single node does not accommodate data confidentiality. This idea made several assumptions, but to list a few, data has been broken into small chunks, and providers cannot lint two chunks of the same data. In the implementation of the botnet's idea, they organized a shared system as a cloud multi-cast service that can be freely accessed. The Onion Routing approach was used When a command is issued from the control node. The control node assumes the function of the botnet, which is responsible for coupling the information and sending it as a response to the request. This requires a high level of trust in the provider; else, there could be a problem.

The resulting key solution to this potential challenge is that all cloud processing providers and cloud storage providers would only get a chunk of the user's data. Each piece of the chunk is prevented from relating with other pieces for a specific user or same data. The providers would not have any idea about the source of each chunk, a better illustration of this is a breakdown puzzle. The idea does not involve encryption of any part of the data, its simple de-aggregation of data, and further keeping the relationship secrete from providers. This improves the confidentiality of data. Two criteria are needed for this idea to work; first, information must be sliced into enough smaller segments to have no meaning. It is claimed that the bigger the segment, the more information it may contain. Secondly, the distribution of these segments must be random across providers to avoid relationships. The transformation needs to be protected by random distribution. Otherwise, a malicious user can easily identify a relationship with less cost.

How would the data be segmented? If the original data is handled like a binary stream and subdivided into smaller streams, then sequential segmentation is the best method.

However, the problem may arise as the number of providers increases. Since the number of the piece is determined by the number of providers, there is a need for an intelligent device, like an agent, that will keep track of the de-fragmentation. Two information is needed to re-assemble the original data; the distribution information allows the process to select the related segment. Information about the order of segments is required to allow accurate reconstruction.

In complex processing, this agent will store every information about the slicing of information and use it for the easy reconstruction of the data to look very much like the original. A test was conducted with an image. Unfortunately, the result shows that if the image is broken into tiny smaller pieces, the reconstruction of the image is different from the original. Another weakness of this idea comes with the update operation on data, which may be difficult if the agent does not keep track of the changes.

A simple solution is to load new data afresh instead of update constantly. Some clients might not trust the providers on assembling the data; an alternative solution is to switch the gear, making the client the work of a provider, re-assembling the data. However, this may require more effort and processing power from the client, but it's a pure solution. This idea seems promising but was difficult to implement because of limited space. The actual splitting of data depends on the type of data, a web API can be used to provide a recipe that dictate the size of blocks of the file and the appropriate number of splits.

3.1.1 Summary Analysis and results of Botnets

The model can still be intruded if dishonest users remove a segment or re-assemble the segment's organization. Suppose an attacker has tremendous access to the service. A disclosed segment can compromise the confidentiality of the original data. Permuting the segments may result in the data in disorder. Fortunately, it requires high computing power and exhaustion to correctly re-assemble the data to its original form. It is not a trivial process to use brute-force search for the entire system. The large storage space that keeps the data will require an equal amount of storage size and a huge cost of network bandwidth to retrieve the data across the network. Results on botnets were provided in terms of delay, throughput, and queue length.

The delay is a measure of time between when the data was sent and when it was received. This delay solely depends on the distance traveled the queue and the processing time. When too many nodes uploads simultaneously, traffic occurs and increases the delay period. Reducing the size, distance, and causes of traffic can consequently reduce delay.

The number of octets received by an agent within the stipulated time is referred to as throughput in this context. If there is a miss of network or transport layer protocol, it will be hampered the packets are dropped. Also, the inbound traffic could affect the amount of throughput; hence increasing the number of nodes yields can increase throughput. More so, Network size is proportional to throughput.

The packet delay is dependent on the queue; the longer the queue, the more the delay. In the evaluation of botnets, there was some notice of network congestion. Hence there is a need to provide an estimate of the queue length. Also, the large packet size reduces the queue to approximately zero.

One big problem with this solution is indexing for search before re-assembling de-fragmented data. This problem is still unsolved; the re-construction has to happen before the search.

3.2 Cryptographic Algorithms

Cryptographic algorithms are used to provide data security; database security [4], and query authentication. This section will summarize algorithms that are presented in work done by Mai Rady et al.

3.2.1 Hash function

The size of data being processed by the local and global network is fast increasing. A hash function is a tool that helps to speed up access and exchanged in a secure manner. A hash function is a one-directional algorithm that takes a lengthy input and outputs a short one, known as a digest. This is also useful to index and retrieve from the database. Examples of algorithms are MD5, which like the internet standard (RFC 1321) that is used to check the integrity of files. SHA1 is another example that uses a hash function to take an input and output a 20-byte hexadecimal number, 40 digits long. It was designed by United States Federal Agency.

3.2.2 Digital signature

This is algorithm involves the user generating both public keys and private keys. A signing function creates the signature, then generates a verification function and provides a message to show that the verification succeeds. Sometimes, the message can be hashed first before the digital signature. This method can effectively protect data, but it is not cost-effective compared to the hash function. There are examples of this algorithm;

RSA (Rivest-Shamir-Adleman signature), each signer has a public key. The message involves computing one modular exponentiation to generate and verify the signature.

DSA (digital signature algorithm), the signer has two keys; the private key is randomly chosen, and the public key is calculated. The message involves computing two modular exponentiation to generate and verify the signature.

BGLS(an aggregate signature scheme) This requires the hash function to map the stream of binary strings to non-zero points in two cyclic groups. BLGS is similar to RSA in terms of the number of keys and modular exponentiation. The three examples above are asymmetric cryptographic, and it requires each message to be validated individually.

3.2.3 Signature aggregation

Digital signature summarized above can be expensive, so this method is to reduce the cost by validating a number of messages with one signature. The aggregation signature is the same as described in the digital signature. The aggregation of a signature that one signer generates is less expensive than the aggregation of signatures signed by different signers. A Signature Aggregation and chaining approach is used to authenticate the query. The server will send matching records with signatures and aggregate them with the condensed-RSA scheme, then chain the signature to produce the correctness and completeness of the query. [6].

3.2.4 Authenticated data structure for integrity

Data integrity can be described in three words; correctness, completeness, and freshness. The integrity at the table level is expensive on a database because the query issuer can only achieve it. Even at the field level, it is complex to sign all fields on the table. Hence, there is overhead in fulfilling data integrity. Data integrity can be accomplished with three entities; cloud service provider, owner/user, and a trusted third party. There are different approaches to provide integrity, such as digital signature, deterministic based scheme, data structure based scheme, and bucket-based index scheme [4]. A better method is the combination of hash function and digital signature schemes. There are static and dynamic scenarios to this; in the static case, the server built a verification object for each query result and paired them to the left and right of the node. And in a dynamic case, the left and right nodes will compute their signature. But the setback of this scheme is the cost of verification. Below is a brief description of common types of authenticated data structure:

MAC(message authentication code) This method produces an authenticator using the private key and input variable-length message. The user will send a query; the algorithm processed it, and the return MAC. However, the algorithm assumes that all users have the same private key.

HMAC(Hash-based message authentication code) This combines the hash function with the private key. It hashes a message and returns digest; then, the user will authenticate the message just like MAC.

MHT(Merkle hash tree) This method fragment data into blocks, then hash each block. The inner nodes and the root can be signed by the user using the private key. It uses the Radix-Path identifier scheme to preserve the order of the blocks. Other examples of authenticated data structure include MBT, which updates the root each time the database is updated (Merkle B-tree), and HLA (public key based homomorphic linear authentication). A log is produced for the verification metadata, and users used it to check the freshness of the query. Each signature has an expiration time, but it is renewed at every update.

3.3 Encryption for confidentiality

In a database, only sensitive data is encrypted, a different part of data can be encrypted with other keys, during retrieval of data, only sensitive data is encrypted or decrypted [11].

Puttaswamy et al. gave a prototype tool, "Silverline" which identifies a subset of data that can be encrypted without affecting functionality. It uses tags to mark data objects and track their usage and dependencies. It encrypts each subset with symmetric encryption with several keys for different users. To fetch the data from the cloud, the user will request the correct key from the owner. The query sent and received by the user will be encrypted then the user decrypts the result. These require the owner's server to be online all the time and hence cause high computation functions to achieve data confidentiality. [8].

Wang et al. proposed a dual encryption method that reexamines the outsourced data. The owner generates primary and secondary keys using a cipher block chaining algorithm. Before retrieving the data, the data is encrypted with the primary key; a subset of the data is encrypted using the secondary key, then merged together and stored in the cloud. This technique an overhead problem on the storage, including the encrypted data and the subset of the data [12].

Omar et al. proposed the use of biometric to encrypt data using voice, fingerprint, and face recognition. The biometric encryption method generates a random key and uses a binding algorithm to merge the biometric image to generate an encrypted key to encrypt data. To decrypt the data, retrieval algorithm, the Biometrically-encrypted key is merged with the biometric image to get the key. And this scheme is useful only with biometric data [7].

4. SECURITY SOLUTIONS COMPARISON

Previous sections summarized and discussed ideas from existing literature and studies on outsourced data confidentiality, integrity, and privacy using different schemes. Data signature is used to ensure data correctness, completeness, and confidentiality when data is encrypted before it is outsourced. The hash algorithm is used to ensure data correctness after it is hashed and signed by the user.

5. DISCUSSION AND PROPOSITION FOR FURTHER WORK

Users tend to be attracted to the privacy part of data security than the confidentiality of data. But most users confused data privacy with data confidentiality. Security assurance is expensive though it is wanted and needed by every data owner. Even if organizations make an effort to provide the mechanisms needed to protect data, most users can not afford or not willing to pay for it. On the other hand, some providers are providing free secured data storage; this has thrown off the need for the user to pay for the same service they can get without paying. For these reasons, most existing solutions are providing barter arrangements for the users in exchange to support the provider's business. In most of the mechanisms discussed, none of them break the need to trust the provider. More so, the schemes in the literature do not discuss whether the identity of users is preserved or not.

Data can be stored in several data copies on many cloud servers; if one server is down as a result of threats, any other server can be used. Hence, this will guarantee the availability of data. Also, there was no standard record of how to determine the appropriate number to slice data in the defragmentation scheme. More so, there should be a level of transparency to whether the customer or the cloud provider would be responsible for a certain security need. This involves the provision of standard policies and legal agreements that capture all ramifications of the security of data.

This paper proposed that the data owner encrypts the data section that passes the sensitivity test using the symmetric encryption algorithm, then sends the authorization to the users. Only authorized users can get the key from the owner using broadcast encryption. This algorithm is

then used to decrypt the result. Hence data confidentiality is certain.
The second part is that the owner can make a Merkle B-Tree data structure on the database, then each attribute has a tree, each value is hashed. The inner node is merged with the root of each tree. Each root is then signed using asymmetric encryption like BGLS. This process guarantees the integrity of the data. A mechanism that isolate virtual machines and associate resources along with best practices of security and regulations at all level.

6. CONCLUSION

Cloud computing has continuously provided data and database services, but the challenges of security never seized. Security is the one challenge that gets the most attention, and it includes confidentiality, availability, integrity, and query processing over encrypted data. This review has a collective summary of common challenges and solutions found in the literature and further provides a proposition that could preserve data confidentiality and integrity.

7. REFERENCES

[1] M. D. Dingledine R., Freedman M. The free haven project: Distributed anonymous storage service. *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[2] G. et al. A quantitative analysis of current security concerns and solutions for cloud computing. *Journal of Cloud Computing: Advances, Systems and Application*, 2012.

[3] K. F. Private virtual infrastructure for cloud computing. *proceedings of the Workshop on Hot Topics in Cloud Computing.*

[4] R. I. Mai Rady, Tamer Abdelkader. Integrity and confidentiality in cloud outsourced data. *Ain Shams Engineering Journal*, 10:275–285, April 2019.

[5] A. V. V. A. A. N. S. A. Martin Gilje Jaatun1*, Gansen Zhao2 and Y. Tang2. The design of a redundant array of independent net-storages for improved confidentiality in cloud computing. *Journal of Cloud Computing.*, 2012.

[6] T. G. Narasimha M. Authentication of outsourced databases using signature aggregation and chaining. *: International conference on database systems for advanced applications (DASFAA)*, 2006.

[7] B. M. Omar MN, Salleh M. Biometric encryption to enhance confidentiality in cloud computing. *International symposium on biometrics and security technologies (ISBAST)*, 2014.

[8] Z. B. Puttaswamy KPN, Kruegel C. Silverline: toward data confidentiality in storage intensive cloud applications. *Symposium on cloud computing (SOCC)*, 2011.

[9] G. D. W. H. Z. B. K. J. Rhea S, Eaton P. Pond: the oceanstore prototype. *Proceedings of the 2nd USENIX Conference on File and Storage Technologies,*, FAST 03, 2003.

[10] S. H. S. S. Ristenpart T, Tromer E. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. *Proceedings of the 16th ACM conference on Computer and communications security*, pages 199–212.

[11] G. E. E. Y. Shmueli E., Vaisenberg R. Implementating a database encryption solution, design and implementation issues. *Elsevier*, 2014.

[12] P. C. Y. P. Wang H, Yin J. Dual encryption for query integrity assurance. *Conference on information and knowledge management*, 2008.

[13] J. M. S. F. Zhao G, Rong C. deployment models for eliminating user concerns on cloud security. *J Supercomputing*, 61(2):337–352, 2012.