Data Security, Privacy, Availability and Integrity in Cloud Computing: Issues and Current Solutions

Sultan Aldossary*
Department of Computer Sciences and cypersecurity
Florida Institute of Technology
Melbourne, Florida 32901

William Allen
Department of Computer Sciences and cypersecurity
Florida Institute of Technology
Melbourne, Florida 32901

*Prince Sattam Bin Abdulaziz University

***Abstract*—Cloud computing changed the world around us. Now people are moving their data to the cloud since data is getting bigger and needs to be accessible from many devices. Therefore, storing the data on the cloud becomes a norm. However, there are many issues that counter data stored in the cloud starting from virtual machine which is the mean to share resources in cloud and ending on cloud storage itself issues. In this paper, we present those issues that are preventing people from adopting the cloud and give a survey on solutions that have been done to minimize risks of these issues. For example, the data stored in the cloud needs to be confidential, preserving integrity and available. Moreover, sharing the data stored in the cloud among many users is still an issue since the cloud service provider is untrustworthy to manage authentication and authorization. In this paper, we list issues related to data stored in cloud storage and solutions to those issues which differ from other papers which focus on cloud as general.**

***Index Terms*—Data security; Data Confidentiality; Data Privacy; Cloud Computing; Cloud Security**

I. Introduction

Cloud computing now is everywhere. In many cases, users are using the cloud without knowing they are using it. According to [1], small and medium organizations will move to cloud computing because it will support fast access to their application and reduce the cost of infrastructure. The Cloud computing is not only a technical solution but also a business model that computing power can be sold and rented. Cloud computing is focused on delivering services. Organization data are being hosted in the cloud. The ownership of data is decreasing while agility and responsiveness are increasing. Organizations now are trying to avoid focusing on IT infrastructure. They need to focus on their business process to increase profitability. Therefore, the importance of cloud computing is increasing, becoming a huge market and receiving much attention from the academic and industrial communities. Cloud computing was defined in [2] by the US National Institute of Standards and Technology (NIST). They defined a cloud computing in [2] as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Schematic definition of cloud computing can be simple, such as seen in Figure 1 1 This

Fig. 1: Schematic definition of cloud computing [3]

cloud model is composed of five essential characteristics, three service models, and four deployment models as in the figure 2. In this technology users outsource their data to a server outside their premises, which is run by a cloud provider [4]. In addition, memory, processor, bandwidth and storage are visualized and can be accessed by a client using the Internet [5]. Cloud computing is composed of many technologies such as service oriented architecture, virtualization, web 2.0 and more. There are many security issues with cloud computing. However, the cloud is needed by organizations due to the need for abundant resources to be used in high demand and the lack of enough resources to satisfy this need. Also, cloud computing offers highly efficient data retrieval and availability. Cloud providers are taking the responsibility of resource optimization.

II. Characteristic of Cloud Computing:

There are five characteristics of cloud computing. The first one is on-demand self-service, where a consumer of services is provided the needed resources without human intervention and interaction with cloud provider. The second characteristic is broad network access, which means resources can be accessed from anywhere through a standard mechanism by thin or thick client platforms such mobile phone, laptop, and desktop computer. Another characteristic is resource pooling, which means the resources are pooled in order for multi-tenants to share the resources. In the multi-tenant model, resources are assigned dynamically to a consumer and after the consumer finishes it, it can be assigned to another one to respond to high resource demand. Even if consumers are assigned to resources on demand, they do not know the location of these

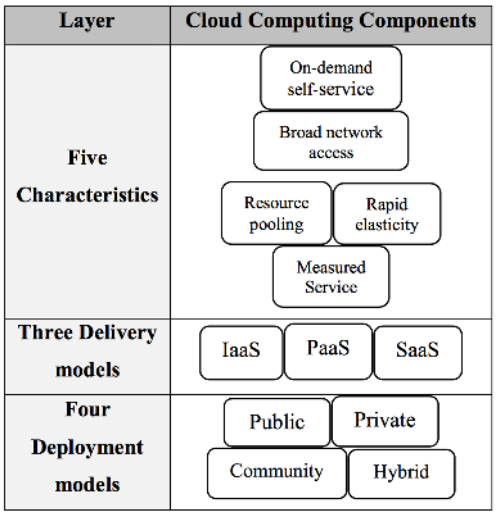

Fig. 2: Cloud environment architecture[6]

assigned resources. Sometimes they know the location at a high-level abstraction, such as country, state, and data center. Storage, processing, memory, and network are the kind of resources that are assigned. Rapid elasticity is also one of the cloud computing characteristics, which means that resources are dynamically increased when needed and decreased when there is no need. Also, one of characteristics that a consumer needs is measured service in order to know how much is consumed. Also, it is needed by the cloud provider in order to know how much the consumer has used in order to bill him or her.

III. Service Models

According to [2], there are three models. Those models differ in the capabilities that are offered to the consumer. It can be software, a platform, or infrastructure. In figure 3, it is comparison between those models with the traditional model.

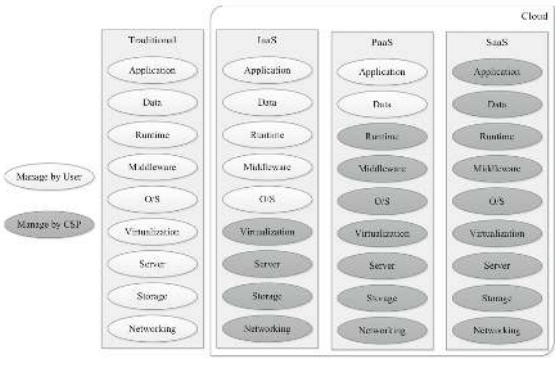

Fig. 3: Service oriented cloud computing architecture[7]

A. Software as a Service (SaaS)

In this service, the cloud service provider provides software and the cloud infrastructure to the clients so they can use this software on the cloud infrastructure for their applications. Since the clients can only run the software and use it, the client does not have control over the underlying infrastructure and physical setting of the cloud such as network, operating system, and storage. The cloud service provider is responsible and is the only one who is in charge of controlling underlying physical setting without client intervention. The client can access this software as a thin client through a web browser.

B. Platform as a Service (PaaS)

This service is similar to SaaS in that the infrastructure is controlled by the cloud service provider but is different in that the users can deploy their software. In this model, the clients can install and deploy their customized applications by using the tool offered by the cloud service provider. Physical settings are controlled and restricted by the cloud service provider and application settings are given to each user to control them.

C. Infrastructure as a Service (IaaS)

In this service, computing resources such as processing, storage and networks can be provisioned. The client of IaaS can install and use any arbitrary operating system. Also, the clients can install and deploy their applications on this operating system. Cloud services such as Amazon EC2 are adopting this model and charging their clients according to the resources are being utilized.

IV. Deployment Models:

Cloud deployment models have been discussed in the literature [8], [9], [10], [11], [12], [13], [14], [15]. There are four deployment models mentioned in [2] as following:

A. Private cloud

In this model, the cloud provider provides cloud infrastructure to a single organization that has many consumers. This infrastructure is to be used exclusively for their use and need. The owner, manager, and operator of this cloud could be the organization itself, a third party, or the organization and third party together. This private cloud could be on premises or off premises.

B. Community Cloud:

In this model, the cloud provider provides cloud infrastructure to many organizations that forms community that shares mission, security requirements, compliance consideration, or policy. this infrastructure is to be used exclusively for their uses and needs. The owner, manager, and operator of this cloud could be one of organizations, a third party, or the organization and third party together. This Community cloud could be on premises or off premises.

C. Public Cloud

This model differs from the previous model in that it is open for the public; it is not private and not exclusively for community. In this model, a public cloud can be provisioned for public to use it to satisfy their needs. The owner, manager,

and operator of this cloud could be a government, private organization, a business or academic organization, and sometimes many of them can be in one cloud and get the service from the same provider.

D. Hybrid Cloud

This model comprises two or more deployment models (private, community, or public). The cloud infrastructure can be combination of those models. Data center within an organization, private cloud, and public cloud can be combined in order to get services and data from both in order to create a well managed and unified computing environment. A cloud can be considered hybrid if the data moves from a data center to a private cloud or public cloud or vice versa.

V. Cloud Security Issues:

Even with these many benefits of cloud computing, previously mentioned, users are reluctant to adopt this technology and move from conventional computing to cloud computing [4]. In cloud computing, security is a broad topic. It is a mix of technologies, controls to safeguard the data, and policies to protect the data, services, and infrastructure. This combination is a target of possible attacks. Therefore, there are new security requirements in the cloud compared to traditional environments. Traditional security architecture is broken because the customer does not own the infrastructure any more. Also, the overall security cloud-based system is equal to the security of the weakest entity [16]. By outsourcing, users lose their physical control over data when it is stored in a remote server and they delegate their control to an untrusted cloud provider or party [17], [18]. Despite powerful and reliable server compared to client processing power and reliability, there are many threats facing the cloud not only from an outsider but also from an insider which can utilize cloud vulnerabilities to do harm [19]. These threats may jeopardize data confidentiality, data integrity, and data availability. Some untrusted providers could hide data breaches to save their reputations or free some space by deleting the less used or accessed data [20].

VI. Top Threats to Cloud Computing

Cloud computing is facing a lot of issues. Those issues are listed as the following: data loss, data breaches, malicious insiders, insecure interfaces and APIs, account or Service hijacking, data location, and denial of Service.

A. Data Loss:

Companies are outsourcing their entire data to cloud service providers. Because of the low cost rate that the cloud offers, the customers should make sure not to expose their important data to risks because of the many ways to compromise their data. In cloud computing, the risks are going up because there are risks that is newly facing the cloud and did not happen to traditional computing, and challenges taking to avoid those risks.[3]. There are many possibilities of losing data due to a malicious attack and sometimes due to server crashes or unintentional deletion by the provider without having backups. Catastrophic events like an earthquake and fire could be the causes of loss. Also, any event that leads to harming the encryption keys could lead to data loss to[21]. In order to avoid losing the data, there are many solutions proposed by CSA[22]:

- Using a strong API for access control
- While the data is in transit, encrypting and protecting its integrity
- Analyzing data protection at run time and design time
- Using strong key generation, storage, destruction, and management practices
- Requiring the service provider to wipe the persistent media data before releasing it to the pool
- Specifying the back up and retention strategies

B. Data Breaches:

A cloud environment has various users and organizations, whose data are in the same place. Any breach to this cloud environment would expose all users' and organizations' data to be unclosed[1]. Because of multi-tenancy, customers using different applications on virtual machines could share the same database and any corruption event that happens to it is going to affect others sharing the same database[21]. Also, even SaaS providers have claimed that they provide more security to customers? data than conventional providers. An insider can access the data but in different ways; he or she is accessing the data indirectly by accessing a lot of information in their cloud and incident could make the cloud insecure and expose customers' data[1]. In [23], it was reported "2011 Data Breach Investigations Report" that hacking and malware are the common causes of data breaches, with 50% hacking and 49% malware.

C. Malicious Insiders:

Malicious insiders are the people who are authorized to manage the data such as database administrators or employees of the company offering cloud services[21], partners, and contractors who have access to the data. Those people can steal or corrupt the data whether they are getting paid by other companies or to just hurt a company. Even the cloud providers may not be aware of that because of their inability in managing their employees. There are many solutions proposed by CSA[22]:

- Conducting a comprehensive supplier assessment and making supply chain management ID stricter
- As part of the legal contract, defining human resources requirements
- Making information security and all cloud service practices more transparent
- creating a process to notify when data breaches happen

D. Insecure interfaces and APIs:

The communication between the cloud service provider and the client is through the API through which the clients can manage and control their data[21]. Therefore, those interfaces

should be secure to prevent any unauthorized access. If they are weak and security mechanism cannot defend them, this could lead to accessing resources even as privileged user. There are many solutions proposed by CSA[22] to avoid insecure interfaces and APIs:

- Analyzing the security model for interfaces of the cloud provider
- Making a strong access control and authentication when data is transmitted
- Understanding dependencies in API

E. Account or Service Hijacking:

Users are using passwords to access the cloud service resources so when their accounts are hijacked and stolen, the passwords are misused and altered unsurprisingly[21]. The unauthorized user who has a password can access the clients' data by stealing it, altering it, or deleting it, or for the benefit of selling it to others. There are many solutions proposed by CSA[22] to avoid account or service hijacking:

- Preventing users from sharing their credentials
- Using a two-factor authentication system
- Monitoring all activities to detect unauthorized access
- Understanding security policies and SLAs

F. Data Location:

Cloud providers have many centers widespread over many places. Data location is an issue in cloud computing since the users of clouds need to know where their data is stored. Some countries, according to jurisdiction, require their companies to store their data in their country. Also, there are regulations in some countries where the company can store their data. Also, the data location matters when the user data is stored in a location that is prone to wars and disasters.

G. Denial of Service:

Some organizations need their systems to be available all the time because availability is important to them due to the critical services they provide. The cloud services provider offers resources that are shared among many clients. If an attacker uses all available resources, others cannot use those resources, which leads to denial of service and could slow accessing those resources. Also, customers, who are using cloud service and affected by botnet, could work to affect availability of other providers.

VII. Multitenancy

In [2], the author did not consider multitenancy as an essential characteristic of cloud computing. However, in CSA [24] and ENISA [25], multi-tenancy is considered an important part of cloud computing. However, with the many benefits multi-tenancy offers, this leads to many challenges regarding having more than one tenant on one physical machine, which is required to utilize the infrastructure. Since tenants are in the same place, they could attack each other. Previously, an attack could be between two separate physical machine but now because two or more tenants are sharing the same hardware, an attacker and a victim can be in the same place. In figure 4, the difference between multi-tenancy and traditional cases is shown. The technology is used to keep tenants from each other by providing a boundary for each tenant by using virtualization. However, virtualization itself is suffering from many issues.

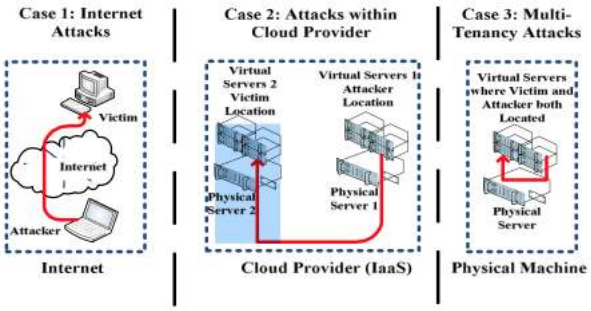

Fig. 4: Difference between Multi-Tenancy and Traditional Cases.[26]

VIII. Virtualization Security Issues

Virtualization is an important component of cloud computing. Now it is getting more attention from academic and industrial communities. Virtualization means separation of underlying hardware resources from provided resources. By using virtualization, two or more operating systems might run in the single machine with each having its own resources.

A. Cross Virtual Machine(VM) Side-Channel Attacks

This attack requires the attacker to be in another virtual machine on the same physical hardware with the victim. In this attack, the attacker and victim are using the same processor and same cache. When the attacker alternates with the victim's VM execution, the attacker can attain some information about the victim's behavior. In [27], there is an example of VM side-channel attack and how the attacker can infer some information about a victim. The timing side channel attack is one kind of VM side channel attacks[28]. This attack is based on determining the time needed by various computations. Determining this time can lead to leaking sensitive information such as described in[28]. This attack can help in leaking some sensitive information such as to the one who performs this computation or sometimes leaking information out of cloud provider itself. This attack is hard to detect because the owner of VM can check other VMs due privacy concern. Sometimes cloud providers can detect a side channel attack but to protect their reputation but they do not announce it. Moreover, there is another type of side channel attacks which is energy-consumption side channel [29].

B. VM Image Sharing

VM can be instantiated from a VM image. A shared image repository can be used to share VM images or a user can have his own VM image [30]. Since there is a repository for sharing VM images, some malicious users could take advantage of this

feature in order to inject a code inside a VM [31]. This will lead to a serious problem. For example, a VM image may contain malware. This malware is coming from the user who used it before[31]. If the image is returned without properly cleaning it, sensitive data could be leaked [30].

C. VM Isolation

Since VMs run in the same hardware, they share all components such as processor, memory, and storage. Isolating of VM logically to prevent one from intervening with another is not enough since they are sharing computation, memory, and storage. Therefore, the data may leak when it is in computation or memory or storage. This is a serious issue. Hence, isolation should be at the level of VM and hardware such as processor, memory, and storage [32].

D. VM escape

The VMs or a malicious user escape from the virtual machine manager(VMM) supervision [33]. VMM controls all VMs and it is the layer that controls how the VM or a user who uses the underlying resources such as hardware. One of the most serious scenarios is that malicious code can go through unnoticed from the VMM and then can interfere with the hypervisor or other guests [31].

E. VM Migration

VM migration process suspends the running VM, copies the status from the source Virtual Machine Monitor (VMM) to the destination VMM and resumes the VM at the destination[11]. In virtual machine migration, the running VM is suspended, has its status copied to the virtual machine monitor (VMM) from its source VMM, and is resumed on the destination VMM[34]. In [35], VM migration is defined as the moving of a VM from one physical machine to another while it is running without shutting it down. Fault tolerance, load balancing, and maintenance are some causes of VM migration [30], [36]. The data and the code of VM [35] are exposed when transferring in the network between two physical hardware locations when they are vulnerable to an attacker. Also, an attacker could let VM transfer to a vulnerable server in order to compromise it. When hen an attacker compromises the VMM, he can get a VM from this data center and migrate it to other centers. Therefore, he can access all resources as a legitimate VM[37]. Therefore, this process incurs more challenge and needs to be secured [30] In order to prevent attackers from benefiting.

F. VM Rollback

This is a process of rolling back a VM to its previous state. Since this process adds more flexibility to the user, it has more security issues. For example, a VM could be rolled back to previous vulnerable state that has not been fixed [38] or it can rolled back to an old security policy or old configuration [30]. In another example, a user could be disabled in a previous state and when the owner of the VM rolls back, the user can still have access [30].

G. Hypervisor Issues:

Hypervisor and virtual machine monitor are the main parts of virtualization. The virtual machine monitor is responsible for managing and isolating VMs from each other. The VMM is the intermediary between the hardware and VMs, so it is responsible for proving, managing, and assigning of the resources. Also, hypervisor with full control of hardware can access Vms' memory[39]. In [39], Jin et al. propose a hardware based soultion to protect VM's memory pages from the malicious hypervisor.

IX. Data Integrity Issues

Data that is stored in the cloud could suffer from the damage on transmitting to/from cloud data storage. Since the data and computation are outsourced to a remote server, the data integrity should be maintained and checked constantly in order to prove that data and computation are intact. Data integrity means data should be kept from unauthorized modification. Any modification to the data should be detected. Computation integrity means that program execution should be as expected and be kept from malware, an insider, or a malicious user that could change the program execution and render an incorrect result. Any deviation from normal computation should be detected. Integrity should be checked at the data level and computation level. Data integrity could help in getting lost data or notifying if there is data manipulation. The following is two examples of how the data integrity could be violated.

A. Data Loss or Manipulation

Users have a huge number of user files. Therefore, cloud providers provide Storage as Service(SasS). Those files can be accessed every day or sometimes rarely. Therefore, there is a strong need to keep them correct. This need is caused by the nature of cloud computing since the data is outsourced to a remote cloud, which is unsecured and unreliable. Since the cloud is untrustworthy, the data might be lost or modified by unauthorized users. In many cases, data could be altered intentional or accidentally. Also, there are many administrative errors that could cause losing data such as getting or restoring incorrect backups. The attacker could utilize the users outsourced data since they have lost the control over it.

B. Untrusted Remote Server Performing Computation on Behave of User

Cloud computing is not just about storage. Also, there are some intensive computations that need cloud processing power in order to perform their tasks. Therefore, users outsource their computations. Since the cloud provider is not in the security boundary and is not transparent to the owner of the tasks, no one will prove whether the computation integrity is intact or not. Sometimes, the cloud provider behaves in such a way that no one will discover a deviation of computation from normal execution. Because the resources have a value to the cloud provider, the cloud provider could not execute the task in a proper manner. Even if the cloud provider is considered more secure, there are many issues such as those coming from

the cloud provider's underlying systems, vulnerable code or misconfiguration.

X. Protecting Data Integrity

Tenants of cloud systems commonly assume that if their data is encrypted before outsourcing it to the cloud, it is secure enough. Although encryption is to provide solid confidentiality against attack from a cloud provider, it does not protect that data from corruption caused by configuration errors and software bugs. There are two traditional ways of proving the integrity of data outsourced in a remote server. Checking the integrity of data can be by a client or by a third party. The first one is downloading the file and then checking the hash value. In this way, a message authentication code algorithm is used. MAC algorithms take two inputs, which are a secret key and variable length of data, which produce one output, which is a MAC (tag). In this way this algorithm is run on the client side. After getting a MAC, the data owner outsources those data to the cloud. For checking its integrity, the data owner downloads the outsourced data and then calculates the MAC for it and compares it with the one calculated before outsourcing that data. By using this method accidental and intentional changes will be detected. Also, by using the key, the authenticity of data will be protected and only the one who has the key can check the data authenticity and integrity. For a large file, downloading and calculating the MAC of the file is an overwhelming process and takes a lot of time. Also, it is not practical since it consumes more bandwidth. Therefore, there is a need for using a lighter technique, which is calculating the hashing value.

The second one is to compute that hash value in the cloud by using a hash tree. In this technique, the hash tree is built from bottom to top where the leaves are the data and parents are also hashed together until the root is reached. The owner of data only stores the root. When the owner needs to check his data, he asks for just root value and compares it with the one he has. This is also to some extent is not practical because computing the hash value of a huge number of values consumes more computation. Sometimes, when the provided service is just storage without computation, the user download the file, the same as in the first case, or send it to third party, which will consume more bandwidth. Therefore, there is a need to find a way to check data integrity while saving bandwidth and computation power. Remote data auditing, by which the data integrity or correctness of remotely stored data is investigated, has been given more attention recently [40], [41], [42], [43], [44], [45]

A. Third Party Auditor

Third Party Auditor (TPA) is the person who has the skills and experience to carry out all auditing processes such as in the figure5. TPA scheme is used for checking the data integrity. Since there are many incidents and doubtful actions, users of cloud storage depend on third party auditors [46]. In [47], Balusamy et al. proposed a framework, which involves the data owner in checking the integrity of their outsourced data.

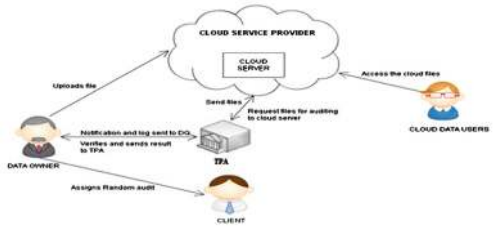

Fig. 5: Architecture of third-party auditing [47]

Their proposed scheme attains data integrity and assures the data owner of the data security. The owner is aware of all his resources on the cloud. Therefore, this scheme guarantees the integrity of data for all owner resources on the cloud. This scheme involves the data owner in the auditing process. First, TPA uses normal auditing processes. Once they discover any modification to the data, the owner is notified about those changes. The owner checks the logs of the auditing process to validate those changes. If the owner suspects that unusual actions have happened to his data, he can check his data by himself or by another auditor assigned by him. Therefore, the owner is always tracking any modification to his own data. There is an assigned threshold value that a response from the third party auditor should not exceed. The data owner validates all modifications lesser than or equal to this threshold. If the time exceeds this threshold, the data owner is supposed to do surprise auditing. The figure 6 shows this auditing process.

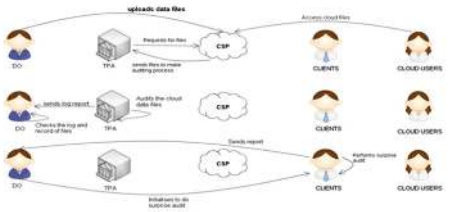

Fig. 6: Proposed scheme architecture [47]

B. Provable Data Possession

In [41] Ateniese et al. proposed the first the Provable Data Possession (PDP) scheme to investigate statically the correctness of the data outsourced to cloud storage without retrieving the data. In [41], the proposed model is to check that data stored in a remote server are still in its possession and that the server has the original data without retrieving it. This model is based on probabilistic proofs by randomly choosing a set of blocks from the server to prove the possession. They used a RSA-based homomorphic verifiable tag, which is combines tags in order to provide a message that the client can use to prove that the server has specific block regardless of whether the client has access to this specific block or not. Even with the advantages this scheme offers, they did not deal with dynamic data storage, and there is computation and communication overhead in the server because of the whole file RSA numbering. In the case of a prover that is untrusted

or has malicious intent, this scheme fails in proofing data possession [7].

In [42], Ateniese et al. overcome the limitation in [41]. By using symmetric cryptography, they proposed a PDP scheme that supports partial and dynamic verification. The limitation of this proposition is that it does not support auditability.

Since PDP schemes just check parts of the file for integrity, there is a need to correct blocks when they suffer from corruption due to hardware issue. In [48], Ateniese et al. propose a scheme to prove data possession with using Forward Error Checking(FEC). First, the file is encoded by using FEC. Then, the encoded file is used by PDP scheme. This methods help in finding the corruptions and mitigating them.

In [44], Wang et al. propose a new dynamic PDP for auditing remote dynamic data. They use the Merkle Hash Function(MHT) and the bilinear aggregate signature. They modify Merkle Hash Function structure by sort leafs node of MHK to be from left to right. This sorting will help in identifying the location of the update. However, this method incur more computation overhead when the file is large.

Sookhak et al.[49] propose a new method for dynamic remote data auditing by using algebraic signature and a new data structure called Divide and Conquer Table(DCT). DCT keep track of the data after appending, updating, insertions, and deletion. Therefore, The need of downloading the file for checking the integrity is avoided.

C. Proof of Retrievability

PDP differs from proof of retrievabilty in that PDP only detects when corruption happens to a large amount of data[50]. PDP protocols can be verified publicly or privately. In the protocol that is privately verifiable, only the owner of the key can verify the encoded data, while in publicly verifiable protocol, data integrity can be verified or audited by a third party. Proof of retrievability is a cryptographic approach based on a challenge response protocol in which a piece of data is proved to be intact and retrievable without retrieving it from the cloud. The the simplest form of proof of retrievability is taking the hash of block using a keyed hash function. Owner of data takes the hash values of the file by using keyed hash function. After getting the hash values, the data owner keep the key and the hash values. the data owner sends the file to a remote server. When the data owner needs to check his data retrievabilty, he sends his key and asks the server to send the hash values by using his key in order to compare them with the hash values that data owner has. The advantage of this solution is that it is simple and implementable. However, there are many disadvantages such that the data owner needs to store many keys in order to use one each time. Also, the number of checking is limited by the number of keys since the remote server could store all keys and the hash values and use them when it is asked to prove having that file. In addition, it costs more resources on the side of a client and server since the hash values need to be calculated each time when the proof is required. Moreover, some thin client such mobile device and PDA does not have the resources to calculate the hash values of big files.

In [50], They used an error correction code and spot checking to prove the possession and retrivability of the data. The verifer hides some sentinels among file blocks before sending them to the remote server. When the verifer wants to check retrivability of the data, it only asks the server for those sentinels. In order to keep those sentinels indistinguishable for the the remote server, the data owner encrypts the file after adding sentinels. In contrast to the simple one, it uses one key regardless of the size of the file. Also, unlike the simple solution that the entire file is processed, it accesses only parts of file. Therefore, the I/O operations is less. This scheme has disadvantages such that the files need to be in encrypted form so it incurs computation overhead in clients such as mobile devices and PDA.

D. Proof of Ownership

In this notion, the client proves ownership of the file outsourced by the client to server. This notion differs from POR and PDP in that POR and PDP need to embed some secret in the file before outsourcing it and the client can check with the cloud server whether the file is in there by asking for the secret and comparing it with what he has. The proof of ownership comes after the need to save some storage by duplication. The owner of the files needs to prove to the server he owns this file.

In [51], Halevi et al. introduced the proof of ownership idea. In [51], the ideas behind proving the ownership are the Collision Resistant Hash functions and Merkle Hash Tree. In [51],The owner of a file creates a Merkle Hash Tree (MHT) and sends the file to the cloud, called verifier. Once it is received by cloud, the file is divided into bits using pairwise independent hash and then the verifier creates a Merkle Hash Tree for this file. Once the prover asks for the ownership of the file, the verifier sends a challenge, which is the root and the number of leaves. The prover calculates the sibling path and returns it to verifier as proof of ownership of this file. The verifier after receiving the sibling path,checks this path against what the merkle tree has and validate the prover. However, this violate the privacy of users since their sensitive data is leaked to the remote server and this issue does not addressed by Halevi et al in [51]. Therefore, there has to be a way to prevent that remote server from accessing outsourced data and building a user profile[52].

XI. Data availability

In [53], Fawaz, et al. developed a storage architecture, figure 7 which covers security, reliability, and availability. The underlying technique of their proposed architecture uses a storage method based on RAID 10. They used three server providers and stripped the data to two servers and the party bits in the third server provider. They followed a sequential way to store the data after encrypting it and dividing the cipher into blocks. One block is in one server provider storage, the next block is in the next server provider storage and the parity

bit in the third server provider. A Parity bit can be in any server provider storage while the other in the other server provider storage. In case the two server providers collide to collect the data, each one has, the encryption will protect the data from unauthorized access. In case one server provider service is distributed, by using a parity bit and an available server provider, the service will be available. Also, it is the same in case one service provider corrupts the data. The number of service provider in this storage architecture can be any number.

In [54], a HAIL (High Availability and integrity Layer) is designed to address the threat caused by a service provider being unavailable. A HAIL distributes the data across many cloud providers to keep their service available all the time. A HAIL leverages many cloud service providers to make a solution that is reliable out of unreliable components and it is cost effective. The idea behind the HAIL is inspired by RAID, which is reliable storage made from unreliable storage. The HAIL works when there is corruption. It does not detect the corruption but it remedies it by avoiding this corruption in a subset of storage providers by using the data in the other service provider storage.

In [55], Bessani et al. proposed Depsky which uses many clouds to build a cloud-of-clouds to address two security requirements in their storage system, which are confidentiality and availability of data. They combined the byzantine quorum protocol as well as secret sharing cryptographic and erasure codes.

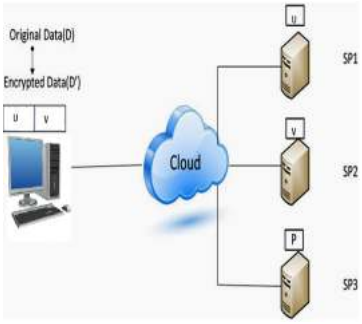

Fig. 7: The proposed parity scheme [53]

XII. Data confidentiality issues

Usually the data is encrypted before it is outsourced. The service provider gets encrypted data. Therefore, it is considered not useful or meaningless. However, the client is responsible for handling the access control policy, encrypting the data, decrypting it and managing the cryptographic keys[56]. Even this would cause a burden to the user; sharing it with others exposes it to risks. When the data is shared among many users, there has to be more flexibility in the encryption process to handle users of the group, manage the keys between users, and enforce the access control policy in order to protect the data confidentiality[57]. Sharing the data among a group of users adds more burden on the owner of the outsourced data.

In [59], the authors describe a cryptosystem in which the data owner encrypts the data by using his public key and

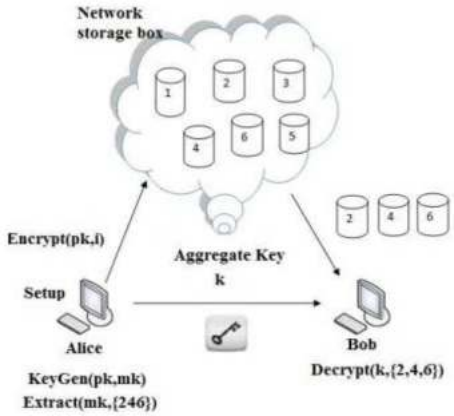

Fig. 8: Key aggregate cryptosystem for sharing data [58]

identifiers called a class on the encryption process. Also, the owner has a master key to create others secret keys for one, some classes of data, or all classes of ciphertext. Once the user gets his aggregate key, he only decrypts the class of ciphertext this key is for. It is an aggregate key where each part of it can decrypt part of the ciphertext.the whole key can decrypt the whole ciphertext. Therefore, this cryptosystem helps in sharing data among a group of users with fine grain access control and without giving them a key that can decrypt all that data. This figure8 shows the general view of this system.

A. Access control:

When data is outsourced to the cloud, which is untrusted because it is in a domain where security is not managed by the data owner, data security has to be given more attention. When more than one entity want to share data, there has to be a mechanism to restrict who can access that data. Many techniques have been discussed in the literature. Those techniques were proposed to keep data content confidential and keep unauthorized entity from accessing and disclosing the data by using access control while permitting many authorized entities to share those data. The following are some of the techniques that are in the literature.

B. Public Key Encryption

Public key encryption is used to encrypt the data by using the public key. Only the one who has the private key can decrypt this data. There are many issues that make this way hard to apply in the cloud when many people need to access those files.

In [60], Sana et el. proposed a lightweight encryption algorithm by utilizing symmetric encryption performance to encrypt files and utilizing asymmetric encryption efficient security to distribute keys. There are many disadvantages of using this method. One of them is key management issue and the need to get fine-grained access to file, such part of it. Also, this solution is not flexible and scalable because encryption and decryption is needed when a user leave the group in order

to prevent him from accessing the data. Key generation and encryption process is shown in figure 9

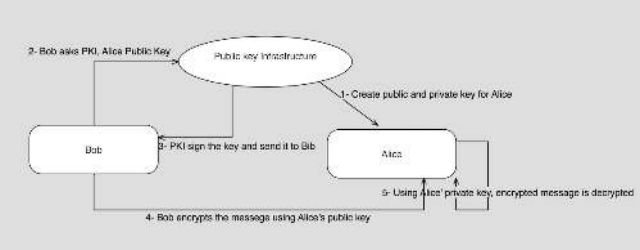

Fig. 9: Public Key Encryption

C. Identity-Based Encryption (IBE)

Shamir, in [61], has introduced identity-based encryption. The owner of data can encrypt his data by specifying the identity of the authorized entity to decrypt it based on that entity?s identity, which must match the one specified by the owner. Therefore, there is no key exchange. Encryption process is shown in figure 10

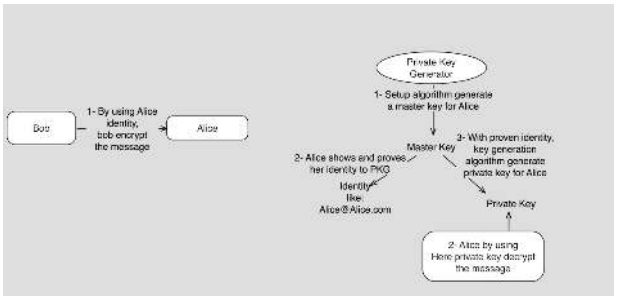

Fig. 10: Identity-Based Encryption (IBE)

D. Attribute Based Encryption (ABE)

In attribute based encryption, an identity of a user is identified by a set of attributes. This set of attributes generates the secret key. Also, it defines the access structure used for access control. This access control are using encryption to encrypt data for confidentiality and share it among group of users. It is a kind of integrating the encryption with the access control.

In [62], attribute-based encryption, know as fuzzy identity-based encryption, was proposed a few years after IBE. In this scheme, a group of attributes identify someone's identity. Data owner encrypts his data and only the one who has attributes that overlap with the attributes specified in the ciphertext can decrypt it. There are general schemes than ABE, which is based on trees. Key generation process is shown in figure 11 and encryption and decryption algorithm is shown in figure 12

1) Key Policy Attribute Based Encryption (KP-ABE): In [63], key policy attribute-based encryption was proposed. This is more general than ABE because it expresses more conditions than just matching the attributes to enforce more

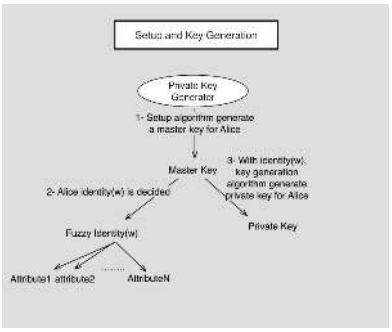

Fig. 11: Attribute Based Encryption (ABE)

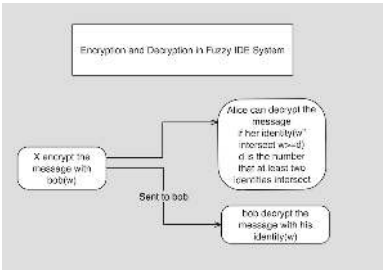

Fig. 12: Encryption\Decryption Attribute Based Encryption (ABE)

control. In this mechanism, ciphertext is linked with a set of attributes. The private key is linked to monotonic access structure. This access structure is based on a tree to specify the identity of the user. When the user?s private key has the attributes that satisfy the attribute in ciphertext, the user decrypts the ciphertext. Key generation process is shown in figure 13 and encryption and decryption algorithm is shown in figure 14. A disadvantage of this method is that the decriptor must trust the key generator to generate keys for a correct person with the right access structure. If the data needs to be re-encrypted, the new private keys have to be issued in order to keep accessing that data. Therefore, there is a need to get the policy associated with the key. Also, it does not support non-monotonic access structure which expresses negative attributes such 'not'.

In [64], Ostrovsky et al. propose a scheme that support non-monotonic access structure which supports positive and negative attributes. However, this scheme increases the size of ciphertext and key. Also, there is cost related to time needed for encryption and decryption. In KP-ABE, the size of ciphertext increases with the number of associated attributes linearly.

In [65], a scheme is proposed that results in constant size of ciphertext regardless of the number of attributes and supports non-monotonic access structure. However, the size of the key is quadratic size of number of the attributes. To overcome that disadvantage, a ciphertext policy attribute-based encryption was proposed. However, CP-ABE costs more than KP-ABE[66].

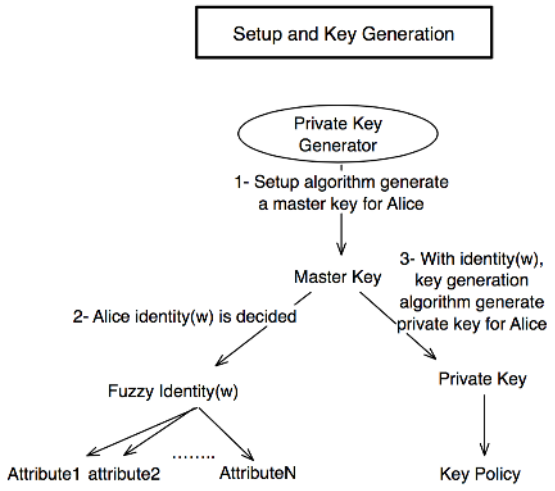

Fig. 13: Key Policy Attribute Based Encryption key Generation

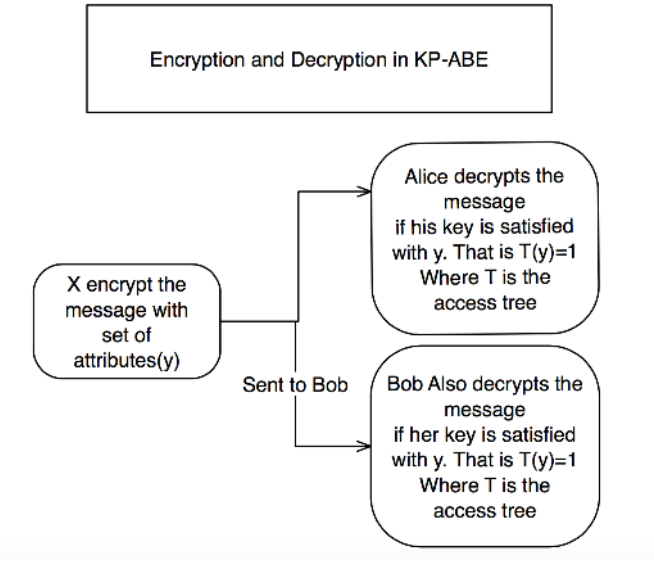

Fig. 14: KP-ABE encryption \decryption

2) Ciphertext Policy Attribute Based Encryption (CP-ABE): In [67], CP-ABE was proposed. In this scheme, the access structure, which is responsible for specifying the encryption policy, is associated with ciphertext. A private key for a user is created based on his attributes. A user can decrypt the ciphertext if the attributes in his private key satisfy the access structure in ciphertext. The benefit of making an access structure with ciphertext is that the encryptor can define the encryption policy and all already-issued private keys can not be changed unless the system is rebooted. There are four functions for the CP-ABE scheme. The four functions are as follows [67][68]. (MasterKey, PublicKey)=Setup(P): A trusted authority runs this function and it takes a security parameter(P)

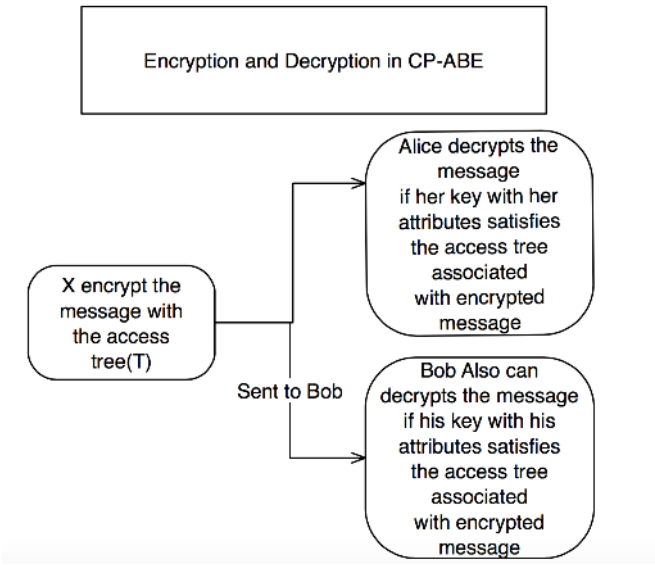

Fig. 15: KP-ABE encryption \decryption

as its input and master key (MK) and public key (PK) as its output.

SK=Key Generation(A,MK): A trusted authority runs this function and it takes a set of attributes (A) and Master Key (MK) as its input and its output is a secret key for a user associated with a set of attributes.

ciphertext (CT)=Encryption (M,MK,P): The data owner runs this function to encrypt his data. It takes a message (M), access control policy (P) and master public key (PK) as its inputs. Its output is a ciphertext under access control policy (P). Encryption algorithm is shown in figure 15

M=Decryption(ciphertext,SK) A decryptor who has the ciphertext runs this function. This ciphertext, under access policy (P) and secret key (SK), can be encrypted if and only if the access policy of the secret key overlap satisfies the access policy of the ciphertext and Its output is the original message. If it does not satisfy those conditions, the decryptor cannot get the original message. decryption algorithm is shown in figure 15.

XIII. Multi-Cloud Computing (MMC) issues

Cloud computing now is moving to multi-cloud computing because of security issues stemming from using a single cloud such data availability. This figure 16 shows how the clients connect to the clouds. Some of the issues that multi-cloud computing are data availability and security [70], Cachinet et al. said ”Services of single clouds are still subject to outage.? There is a fear among organizations that a single cloud would not fulfill their demands such as reliability and availability. Some organizations need the availability to be high and need their data to be far from being locked in. Therefore, they need a system that is always available and not under control of a single cloud provider. The notion of a multi-cloud will be become a trend in these years.

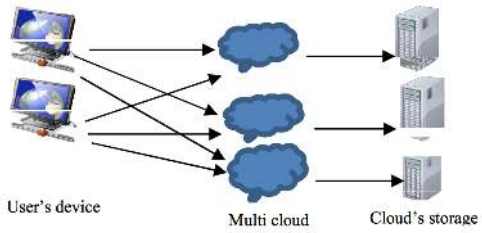

Fig. 16: Multi-cloud computing [69]

In [6], Alzain et al. have discussed many security issues in a single cloud and they are promoting the multi-cloud and its solutions to address single cloud security issues. They promised by using multi-cloud, valuable information such as credit card information and medical records could be protected from untrusted third parties and malicious insiders.

In [71], the authors said that moving from a single cloud to multi-cloud distributes trust, reliability, and security among multiple cloud providers. In addition to that, the users can avoid moving their data once they got locked in, by using another clouds to run their business.

In [72], Mahesh et al. suggests encrypting data, dividing it into chunks and storing those chunks in many cloud service providers. They insisted this would help to prevent all security issues of the cloud.

In [73], SUGANTHI et al. proposed a solution for protecting the privacy of the signer of that data from a third party auditor while auditing process. When an owner of data partions their data and sign them and distribute them to multi-clouds and share them with others, the third party could get the identity of the signer since it is needed when auditing. Therefore, they proposed this solution to prevent violating the privacy of the owner by knowing their identity by using creating homomorphic authenticators by using aggregate signatures[73]. Aggregate signature scheme is a group of signatures that are aggregated to one digital signature[74]. One Aggregate signature for n signatures of m messages that are from u users is the result of this scheme[74]. Therefore, the benefit of using it here is that the auditor will know the users how sign the messages but without knowing specifically how sign each message.

XIV. Mobile Cloud Computing

A. Limitations of mobile devices

With the advancement in mobile devices such as more processing, storage, memory, sensors and operating system capabilities, there is a limitation with regard to energy resources needed for complex computation. Some of the application in mobile devices are data-intensive or compute-intensive application. Due to battery life, the mobile device cannot run them. Therefore, the cloud computing is needed to run those complex computations. The mobile device's application augments the processing tasks to the cloud computing.

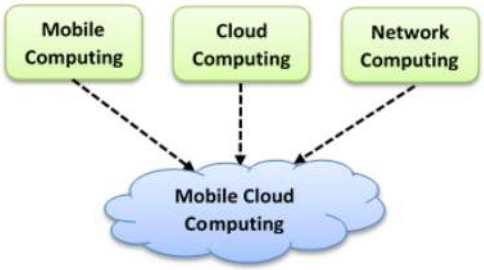

Figure 1. Evolution of Mobile Cloud Computing

Fig. 17: Mobile cloud computing [75]

B. Mobile Cloud Computing

Mobile cloud computing is using the mobile as front end and the cloud as back end for the storage and computation. In the figure 17, mobile cloud computing consists of mobile computing, cloud computing, and network.

In [76], three schemes are proposed for confidentiality and integrity of mobile device's files stored in the cloud. The first scheme is encryption based Scheme(EnS). In this scheme, the mobile device encrypts the file and gets its hash code. The encryption key is a concatenation of the password entered by a user, file name changed to bits and file size to defend brute force attack on a cloud server since the length of the password is limited. Only the file name is kept in the file and everything related to the file is deleted. When downloading the file from the cloud server, only the password is needed to decrypt the file. This process will need more processing on the mobile device side. They proved the confidentiality and integrity of the file using this scheme when it is stored in a distrusted clouds server. In order to overcome the power consumption in the first scheme, a coding based scheme is proposed. This scheme is not using encryption function since it consumes less power. The confidentiality of the file is protected by using matrix multiplication and the integrity is ensured by using hash-based message authentication code. The file is divided to many blocks and each block is divided to many chunks and each chunk in n bits. Each block represents matrix with chunks number as rows and bits as columns. a code victor matrix is created from the entered password. For confidentiality, each matrices are multiplied by the code victor matrix which result in secrecy code. For the integrity, all secrecy codes are concatenated and hashed. The result of the previous is the integrity key. The file is hashed with the integrity key which results in message authentication code. The third scheme is Sharing based Scheme(ShS) which applies X-OR operations on the file. This scheme needs less computational power. Hash-based message authentication code is used to verify the integrity of file while X-or operation is used to protect the confidentiality of the file.

In [77], Khan et al. propose a new scheme called block-based sharing scheme. This scheme overcomes all limitations of the previous schemes proposed in [76]. They use X-OR operation. First, they extend the password entered by a user

in order to be the same as block size. For example, the block size is 160 bit and the password entered by the user is 60 bits. In this case, they extend 60 bits to be 160 bits. Second, they divide a file to blocks with the same size. After that, they X-or the first block with first extended password. The second block is X-ORed with extended password after shifting each bit to the right. Therefore, each block is x-ORed with distinct password with size equal to the size of block. For integrity, they hash the concatenation of the file name, extended password and file size in order to get an integrity key. Then, they hash the file with the integrity key in order to get message authentication code. Once that done, only cipher text, message authentication code, and the hash of file name to the cloud. The hash of file name is sent for file retrieval. This scheme results in less energy consumption, memory utilization, and CPU utilization.

In [78], the authors used homomorphic encryption, multi-cloud computing and mobile. They used multiple cloud schemes for storing the data to avoid data lock in and used homomorphic encryption to run computations without downloading the data back and forth between cloud computing and mobile to avoid the communication costs. Since encryption is expensive for the mobile devices, there are some propositions to avoid using it.

In [79], Bahrami1 et al. proposed a lightweight method for data privacy in mobile cloud computing. They used JEPG file as their case study because it is a common file in mobile. They divide the JEPG file into many splits, distribute them to many file based one predefined pattern, and scramble chunks randomly in each split file with help of psedue-random permutations with the chaos system. After that each file is sent to MCCs. For retrieval process, the split files are collected from MCCs. Each split chunks are rearranged by using the chaos system. After that, all split files is rearranged based pattern, predefined before. They used this method because it is low in computation and works effectively in the mobile. When they compared it with encrypting the JEPG in the mobile and sending it, they found their solution is more efficient. Their proposed method has two requirements: balancing computation overhead with maintaining the security and avoiding offloading the file to the mobile cloud computing for encryption by making the file is meaningless before sending it.

XV. Conclusion

Cloud computing is an emerging technology that will receive more attention in the future from industry and academia. The cost of this technology is more attractive when it is compared to building the infrastructure. However, there are many security issues coming with this technology as happens when every technology matures. Those issues include issues related to the previous issues of the internet, network issues, application issues, and storage issues. Storing data in a remote server leads to some security issues. Those issues are related to confidentiality of data from unauthorized people in remote sites, integrity of stored data in remote servers and the availability of the data when it is needed. Also, sharing data in cloud when the cloud service provider is mistrusted is an issue. However, we mentioned some techniques that protect data seen by the cloud service provider while it is shared among many users.Many studies have been conducted to discover the issues that affect confidentiality, integrity, and availability of data to find a solution for them. Those solutions will lead to more secure cloud storage, which will also lead to more acceptance from the people and the trust on the cloud will increase.

References

[1] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1 – 11, 2011.

[2] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

[3] M. T. Khorshed, A. S. Ali, and S. A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," *Future Generation Computer Systems*, vol. 28, no. 6, pp. 833 – 851, 2012.

[4] Z. Zhou and D. Huang, "Efficient and secure data storage operations for mobile cloud computing," in *Proceedings of the 8th International Conference on Network and Service Management*. International Federation for Information Processing, 2012, pp. 37–45.

[5] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, no. 4, pp. 51–56, 2010.

[6] 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.

[7] M. Sookhak, H. Talebian, E. Ahmed, A. Gani, and M. K. Khan, "A review on remote data auditing in single cloud server: Taxonomy and open issues," *Journal of Network and Computer Applications*, vol. 43, pp. 121–141, 2014.

[8] E. Aguiar, Y. Zhang, and M. Blanton, "An overview of issues and recent developments in cloud computing and storage security," in *High Performance Cloud Auditing and Applications*. Springer, 2014, pp. 3–33.

[9] I. Gul, M. Islam *et al.*, "Cloud computing security auditing," in *Next Generation Information Technology (ICNIT), 2011 The 2nd International Conference on*. IEEE, 2011, pp. 143–148.

[10] E. M. Mohamed, H. S. Abdelkader, and S. El-Etriby, "Enhanced data security model for cloud computing," in *Informatics and Systems (INFOS), 2012 8th International Conference on*. IEEE, 2012, pp. CC–12.

[11] S. Ramgovind, M. M. Eloff, and E. Smith, "The management of security in cloud computing," in *Information Security for South Africa (ISSA), 2010*. IEEE, 2010, pp. 1–7.

[12] F. Sabahi, "Cloud computing security threats and responses," in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*. IEEE, 2011, pp. 245–249.

[13] X. Wang, B. Wang, and J. Huang, "Cloud computing and its key techniques," in *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011, pp. 404–410.

[14] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of network and computer applications*, vol. 34, no. 1, pp. 1–11, 2011.

[15] J. Yang and Z. Chen, "Cloud computing research and security issues," in *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*. IEEE, 2010, pp. 1–3.

[16] M. Lori, "Data security in the world of cloud computing," *Co-published by the IEEE Computer And reliability Societies*, pp. 61–64, 2009.

[17] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," *Network, IEEE*, vol. 24, no. 4, pp. 19–24, 2010.

[18] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos, "Security and privacy for storage and computation in cloud computing," *Information Sciences*, vol. 258, pp. 371–386, 2014.

[19] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *INFOCOM, 2010 Proceedings IEEE*. Ieee, 2010, pp. 1–9.

[20] K. Yang and X. Jia, "Data storage auditing service in cloud computing: challenges, methods and opportunities," *World Wide Web*, vol. 15, no. 4, pp. 409–428, 2012.

[21] CSA, "The notorious nine cloud computing top threats in 2013," $The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf$.

[22] D. Hubbard and M. Sutton, "Top threats to cloud computing v1. 0," *Cloud Security Alliance*, 2010.

[23] W. Baker, "M," 2011 data breach investigations report,"." [Online]. Available: http://www.wired.com/images$_blogs/threatlevel/2011/04/Verizon$ $2011-DBIR_04-13-11.pdf$

[24] G. Brunette, R. Mogull *et al.*, "Security guidance for critical areas of focus in cloud computing v2. 1," *Cloud Security Alliance*, pp. 1–76, 2009.

[25] D. Catteddu, "Cloud computing: benefits, risks and recommendations for information security," in *Web Application Security*. Springer, 2010, pp. 17–17.

[26] H. Aljahdali, A. Albatli, P. Garraghan, P. Townend, L. Lau, and J. Xu, "Multi-tenancy in cloud computing," in *Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on*, April 2014, pp. 344–351.

[27] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 199–212.

[28] A. Aviram, S. Hu, B. Ford, and R. Gummadi, "Determinating timing channels in compute clouds," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*. ACM, 2010, pp. 103–108.

[29] H. Hlavacs, T. Treutner, J.-P. Gelas, L. Lefevre, and A.-C. Orgerie, "Energy consumption side-channel attack at virtual machines in a cloud," in *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*. IEEE, 2011, pp. 605–612.

[30] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing," *Journal of Internet Services and Applications*, vol. 4, no. 1, pp. 1–13, 2013.

[31] W. A. Jansen, "Cloud hooks: Security and privacy issues in cloud computing," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*. IEEE, 2011, pp. 1–10.

[32] N. Gonzalez, C. Miers, F. Redígolo, M. Simplicio, T. Carvalho, M. Näslund, and M. Pourzandi, "A quantitative analysis of current security concerns and solutions for cloud computing," *Journal of Cloud Computing*, vol. 1, no. 1, pp. 1–18, 2012.

[33] M. H. Song, "Analysis of risks for virtualization technology," in *Applied Mechanics and Materials*, vol. 539. Trans Tech Publ, 2014, pp. 374–377.

[34] R. Bifulco, R. Canonico, G. Ventre, and V. Manetti, "Transparent migration of virtual infrastructures in large datacenters for cloud computing," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*. IEEE, 2011, pp. 179–184.

[35] F. Zhang and H. Chen, "Security-preserving live migration of virtual machines in the cloud," *Journal of network and systems management*, vol. 21, no. 4, pp. 562–587, 2013.

[36] A. Corradi, M. Fanelli, and L. Foschini, "Vm consolidation: A real case based on openstack cloud," *Future Generation Computer Systems*, vol. 32, pp. 118–127, 2014.

[37] S. Fiebig, M. Siebenhaar, C. Gottron, and R. Steinmetz, "Detecting vm live migration using a hybrid external approach." in *CLOSER*, 2013, pp. 483–488.

[38] H. Wu, Y. Ding, C. Winer, and L. Yao, "Network security for virtual machine in cloud computing," in *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*. IEEE, 2010, pp. 18–21.

[39] S. Jin, J. Ahn, S. Cha, and J. Huh, "Architectural support for secure virtualization under a vulnerable hypervisor," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2011, pp. 272–283.

[40] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proceedings of the 16th ACM conference on Computer and communications security*. Acm, 2009, pp. 213–222.

[41] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM conference on Computer and communications security*. Acm, 2007, pp. 598–609.

[42] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*. ACM, 2008, p. 9.

[43] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 9, pp. 1717–1726, 2013.

[44] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 5, pp. 847–859, 2011.

[45] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," *Services Computing, IEEE Transactions on*, vol. 5, no. 2, pp. 220–232, 2012.

[46] C. Wang, S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *Computers, IEEE Transactions on*, vol. 62, no. 2, pp. 362–375, Feb 2013.

[47] B. Balusamy, P. Venkatakrishna, A. Vaidhyanathan, M. Ravikumar, and N. Devi Munisamy, "Enhanced security framework for data integrity using third-party auditing in the cloud system," in *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, ser. Advances in Intelligent Systems and Computing. Springer India, 2015, vol. 325, pp. 25–31.

[48] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote data checking using provable data possession," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 12:1–12:34, Jun. 2011.

[49] M. Sookhak, A. Gani, M. K. Khan, and R. Buyya, "Dynamic remote data auditing for securing big data storage in cloud computing," *Information Sciences*, 2015.

[50] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in *Proceedings of the 14th ACM conference on Computer and communications security*. Acm, 2007, pp. 584–597.

[51] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 491–500.

[52] N. Kaaniche and M. Laurent, "A secure client side deduplication scheme in cloud storage environments," in *New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on*, March 2014, pp. 1–7.

[53] F. S. Al-Anzi, A. A. Salman, N. K. Jacob, and J. Soni, "Towards robust, scalable and secure network storage in cloud computing," in *Digital Information and Communication Technology and it's Applications (DICTAP), 2014 Fourth International Conference on*. IEEE, 2014, pp. 51–55.

[54] K. D. Bowers, A. Juels, and A. Oprea, "Hail: a high-availability and integrity layer for cloud storage," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 187–198.

[55] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "Depsky: dependable and secure storage in a cloud-of-clouds," *ACM Transactions on Storage (TOS)*, vol. 9, no. 4, p. 12, 2013.

[56] D. Chen, X. Li, L. Wang, S. Khan, J. Wang, K. Zeng, and C. Cai, "Fast and scalable multi-way analysis of massive neural data," *IEEE Trans. Comput*, vol. 63, 2014.

[57] A. N. Khan, M. M. Kiah, S. A. Madani, M. Ali, S. Shamshirband *et al.*, "Incremental proxy re-encryption scheme for mobile cloud computing environment," *The Journal of Supercomputing*, vol. 68, no. 2, pp. 624–651, 2014.

[58] P. S. Kumari, P. Venkateswarlu, and M. Afzal, "A key aggregate framework with adaptable offering of information in cloud," *International Journal of Research*, vol. 2, no. 3, pp. 5–10, 2015.

[59] C.-K. Chu, S. S. Chow, W.-G. Tzeng, J. Zhou, and R. H. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 468–477, 2014.

[60] R. A. Sana Belguith, Abderrazak Jemai, "Enhancing data security in cloud computing using a lightweight cryptographic algorithm," *ICAS 2015 : The Eleventh International Conference on Autonomic and Autonomous Systems*, 2015.

[61] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in cryptology*. Springer, 1985, pp. 47–53.

[62] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*. Springer, 2005, pp. 457–473.

[63] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*. Acm, 2006, pp. 89–98.

[64] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 195–203.

[65] N. Attrapadung, B. Libert, and E. De Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Public Key Cryptography–PKC 2011*. Springer, 2011, pp. 90–108.

[66] Z. Qiao, S. Liang, S. Davis, and H. Jiang, "Survey of attribute based encryption," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014 15th IEEE/ACIS International Conference on*, June 2014, pp. 1–6.

[67] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007, pp. 321–334.

[68] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 7, pp. 1214–1221, 2011.

[69] H. Hasan and S. Chuprat, "Secured data partitioning in multi cloud environment," in *Information and Communication Technologies (WICT), 2014 Fourth World Congress on*, 2014, pp. 146–151.

[70] C. Cachin, I. Keidar, and A. Shraer, "Trusting the cloud," *ACM SIGACT News*, vol. 40, no. 2, pp. 81–86, 2009.

[71] M. Vukolić, "The byzantine empire in the intercloud," *ACM SIGACT News*, vol. 41, no. 3, pp. 105–111, 2010.

[72] M. Shankarwar and A. Pawar, "Security and privacy in cloud computing: A survey," in *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, ser. Advances in Intelligent Systems and Computing. Springer International Publishing, 2015, vol. 328, pp. 1–11.

[73] J. Suganthi, J. Ananthi, and S. Archana, "Privacy preservation and public auditing for cloud data using ass in multi-cloud," in *Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on*, 2015, pp. 1–6.

[74] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Advances in cryptologyEUROCRYPT 2003*. Springer, 2003, pp. 416–432.

[75] A. Donald and L. Arockiam, "A secure authentication scheme for mobicloud," in *Computer Communication and Informatics (ICCCI), 2015 International Conference on*, Jan 2015, pp. 1–6.

[76] W. Ren, L. Yu, R. Gao, and F. Xiong, "Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing," *Tsinghua Science & Technology*, vol. 16, no. 5, pp. 520–528, 2011.

[77] A. N. Khan, M. M. Kiah, M. Ali, S. A. Madani, S. Shamshirband *et al.*, "Bss: block-based sharing scheme for secure data storage services in mobile cloud environment," *The Journal of Supercomputing*, vol. 70, no. 2, pp. 946–976, 2014.

[78] 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.

[79] M. Bahrami and M. Singhal, "A light-weight permutation based method for data privacy in mobile cloud computing," in *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 3rd IEEE International Conference on*, March 2015, pp. 189–198.