A Survey on Data Security Issues in Cloud Computing: From Single to Multi-Clouds

Mohammed A. AlZain, Ben Soh and Eric Pardede
Department of Computer Science and Computer Engineering,
La Trobe University, Bundoora 3086, Australia.
Email: [maalzain@students., B.soh@, E.Pardede@]latrobe.edu.au

***Abstract*—Cloud computing usage has increased rapidly in many companies. Cloud computing offers many benefits in terms of low cost and accessibility of data. Ensuring the security of cloud computing plays a major role in the cloud computing, as customers often store important information with cloud storage providers but these providers may be unsafe. Customers are wondering about attacks on the integrity and the availability of their data in the cloud from malicious insiders and outsiders, and from any collateral damage of cloud services. These issues are extremely significant but there is still much room for security research in cloud computing. Dealing with "single cloud" providers is predicted to become less popular with customers due to risks of service availability failure and the possibility of malicious insiders in the single cloud. A movement towards "multi-clouds", or in other words, "interclouds" or "cloud-of-clouds" has increased recently.**

The purpose of this paper is to survey recent research related to single and multi-clouds security and to address possible solutions. It is found that the research into the use of multi-cloud providers to maintain security has received less attention from the research community than has the use of single clouds. This work aspires to promote the use of multi-clouds due to its ability to reduce security risks that affect the cloud computing consumer.

***Index Terms*—Cloud computing, single cloud, multi-clouds, data confidentiality, data integrity, service availability.**

I. INTRODUCTION

The use of cloud computing has improved rapidly in many organizations. Subashini and Kavitha [61] argue that small and medium companies use cloud computing services for various reasons, including providing fast access to their applications and reducing their infrastructure costs. The cloud computing projects spending in USA between 2010 to 2015 will be at 40% annual growth rate (CAGR) and will pass $7 billion by 2015 [36]. In addition, researchers estimated that 12% of software market will move toward cloud computing within the next 5 years and the amount growth of cloud computing market will reach $95 billion [61].

Security and privacy are considered to be the critical aspects in a cloud computing environment due to the sensitive and important information stored in the cloud for customers. Cloud providers should address security and privacy issues as a matter of high and urgent priority.

Dealing with "single cloud" providers is becoming less popular with customers due to potential problems such as service availability failure and the possibility that there are malicious insiders in the single cloud. In recent years, there has been a move towards "multi-clouds", "intercloud" or "cloud-of-clouds".

This paper focuses more on the issues related to the data security and privacy aspects in cloud computing, such as data integrity, data confidentiality and service availability. As data and information will be shared with a third party, cloud computing customers want to avoid an unsafe or untrusted cloud provider. Protecting private and important information, such as credit card details or a patient's medical records from attackers or malicious insiders is of critical importance. In addition, the potential for migration from a single cloud to a multi-cloud environment is examined and research related to security issues in single and multi-clouds in cloud computing is surveyed.

The remainder of this paper is organized as follows. Section II describes the beginning of cloud computing and its components. In addition, it presents examples of cloud providers and the benefits of using their services. Section III discusses security risks in cloud computing. Section IV analyses the new generation of cloud computing, that is, multi-clouds and recent solutions to address the security of cloud computing, as well as examining their limitations. Section V presents suggestions for future work. Section VI will conclude the paper.

II. CLOUD COMPUTING: PRELIMINARY

According to [15],[30] ,[68], cloud computing has been defined as ''a style of computing where massively scalable IT-enabled capabilities are delivered 'as a service' to external customers using Internet technologies''. Moreover, NIST [1] describes cloud computing as "a model for enabling 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".

doi:10.4304/jsw.8.5.1068-1078

A. *Cloud Computing Components and Layers*

The cloud computing model consists of five characteristics, three delivery models, and four deployment models [1]. The five key characteristics of cloud computing are: location-independent resource pooling, on-demand self-service, rapid elasticity, broad network access, and measured service [63]. These five characteristics represent the first layer in the cloud environment architecture (see Figure 1).

Layer	Cloud Computing Components
Five Characteristics	On-demand self-service; Broad network access; Resource pooling; Rapid elasticity; Measured Service
Three Delivery models	IaaS; PaaS; SaaS
Four Deployment models	Public; Private; Community; Hybrid

Figure 1: Cloud Environment Architecture-adopted from NIST definition [1].

The three key cloud delivery models are infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). In IaaS, the user can benefit from networking infrastructure facilities, data storage and computing services. In other words, it is the delivery of computer infrastructure as a service. An example of IaaS is the Amazon web service [35]. In PaaS, the user runs custom applications using the service provider's resources. It is the delivery of a computing platform and solution as a service. An example of PaaS is GoogleApps [35]. Running software on the provider's infrastructure and providing licensed applications to users to use services is known as SaaS. An example of SaaS is the Salesforce.com CRM application [35],[61],[63]. This model represents the second layer in the cloud environment architecture.

Cloud deployment models include public, private, community, and hybrid clouds. A cloud environment that is accessible for multi-tenants and is available to the public is called a public cloud. A private cloud is available for a particular group, while a community cloud is modified for a specific group of customers. Hybrid cloud infrastructure is a composition of two or more clouds (private, community, or public cloud) [63]. This model represents the third layer in the cloud environment architecture.

Kamara and Lauter [35] present two types of cloud infrastructure only, namely private and public clouds. The infrastructure that is owned and managed by users is in the private cloud. Data that is accessed and controlled by trusted users is in a safe and secure private cloud, whereas the infrastructure that is managed and controlled by the cloud service provider is in a public cloud. In particular, this data is out of the user's control, and is managed and shared with unsafe and untrusted servers [35].

B. *Cloud Service Providers: Single Cloud*

There are many types of cloud computing. First, cloud storages, such as Amazon S3, Microsoft SkyDrive, or Nirvanix CLoudNAS [18], permit consumers to access online data. Second, it provides computation resources for users such as Amazon EC2. Third, it provides online collaboration facilities such as Google Apps or versioning repositories for source code [18].

Cloud service providers should ensure the security of their customers' data and should be responsible if any security risk affects their customers' service infrastructure. A cloud provider offers many services that can benefit its customers, such as fast access to their data from any location, scalability, pay-for-use, data storage, data recovery, protection against hackers, on-demand security controls, and use of the network and infrastructure facilities [61].

One of the first cloud computing implementations to deliver project services through a website was introduced by Salesforce.com in 1999 [4]. Amazon Web Services in 2002 provided customers with advantages such as storage and computation services. In 2006, Amazon provided their customers with the Elastic Compute Cloud (EC2) service to allow them to use their instance for data processing and computing [36]. Amazon produced the Amazon Elastic compute Cloud (EC2) as a cloud service [9] to allow users to purchase computational resources, without the need to have significant technical background to deal with the cloud computing environment. Users can focus on their own applications instead of maintaining the cloud environment software and hardware. Amazon EC2 is a cloud service that offers virtual machine instances on-demand which provides users with a super computer equivalent without the need to purchase it. The cost of renting the services of a cloud service provider (as-you-go) is cheaper than purchasing a super computer for the same purpose [4].

Amazon EC2 also supports the building of a cloud computing environment as a group of virtual machines technology. Supporters of this technology argue that software and hardware for this computational environment are compatible with each other and users do not need to perform compatibility measures between them. Opponents of this methodology argue about the overheads of virtual machines and the negative aspects of sharing one physical machine with other virtual machines [4].

In the cloud computing environment, Amazon EC2 is a collection of virtual machine nodes or instance. In relation to the user's charges for Amazon EC2, the communication between instances and communication between instances and machines outside Amazon EC2 will be charged based on CPU time [9]. Kaufman [36] developed a security model that ensures data confidentially, data integrity, and data availability (CIA). The cloud storage provider must be able to provide an encryption schema for the stored data, access control for their data to prevent an unauthorized user from accessing the data, and provide a backup service for their data.

Public cloud services for data storage, such as Amazon Simple Storage Service (S3) and Azure in Microsoft, provide customers with dynamic and scalable storage services. The public cloud, as discussed before, protects the user from the cost of purchasing hardware and software for their storage infrastructure; instead, they pay a cloud service provider [4].

Reliability and availability are other benefits for the public cloud, in addition to low cost [35]. However, there are also concerning issues for public cloud computing, most notably, issues surrounding data integrity and data confidentiality.

III. SECURITY ISSUES

Although cloud service providers can provide benefits to consumers, security risks play a major role in the cloud computing environment [65]. Users of online data sharing or network facilities are aware of the potential loss of privacy [18]. According to a recent IDC survey [23], the top challenge for 74% of CIOs in relation to cloud computing is security. Protecting private and important information such as credit card details or patients' medical records from attackers or malicious insiders is of critical importance [45]. Moving databases to large data centers involves many security challenges [67] such as virtualization vulnerability, accessibility vulnerability, privacy and control issues related to data accessed from a third party, integrity, confidentiality, and data loss or theft. Subashini and Kavitha [61] present some fundamental security challenges, which are data storage security, application security, data transmission security, and security related to third-party resources.

In different cloud service models, the security responsibility between users and providers is different. According to Amazon [58], their EC2 addresses security control in relation to physical, environmental, and virtualization security, whereas, the users remain responsible for addressing security control of the IT system including the operating systems, applications and data.

According to Takabi et al. [63], the way the responsibility for privacy and security in a cloud computing environment is shared between consumers and cloud service providers differs between delivery models. In SaaS, cloud providers are more responsible for the security and privacy of application services than the users. This responsibility is more relevant to the public than the private cloud environment because the clients need more strict security requirements in the public cloud. In PaaS, users are responsible for taking care of the applications that they build and run on the platform, while cloud providers are responsible for protecting one user's applications from others. In IaaS, users are responsible for protecting operating systems and applications, whereas cloud providers must provide protection for the users' data [63].

Ristenpart et al. [53] claim that the levels of security issues in IaaS are different. The impact of security issues in the public cloud is greater than the impact in the private cloud. For instance, any damage which occurs to the security of the physical infrastructure or any failure in relation to the management of the security of the infrastructure will cause many problems. In the cloud environment, the physical infrastructure that is responsible for data processing and data storage can be affected by a security risk. In addition, the path for the transmitted data can be also affected, especially when the data is transmitted to many third-party infrastructure devices [53].

As the cloud services have been built over the Internet, any issue that is related to internet security will also affect cloud services. Resources in the cloud are accessed through the Internet; consequently even if the cloud provider focuses on security in the cloud infrastructure, the data is still transmitted to the users through networks which may be insecure. As a result, internet security problems will affect the cloud, with greater risks due to valuable resources stored within the cloud and cloud vulnerability. The technology used in the cloud is similar to the technology used in the Internet. Encryption techniques and secure protocols are not sufficient to protect data transmission in the cloud. Data confidentiality of the cloud through the Internet by hackers and cybercriminals needs to be addressed and the cloud environment needs to be secure and private for clients [61].

We will address three security factors that particularly affect single clouds, namely data integrity, data confidentiality, and service availability.

A. *Data Integrity*

One of the most important issues related to cloud security risks is data integrity. The data stored in the cloud may suffer from damage during transition operations from or to the cloud storage provider. Cachin et al. [18] give examples of the risk of attacks from both inside and outside the cloud provider, such as the recently attacked Red Hat Linux's distribution servers [52]. Another example of breached data occurred in 2009 in Google Docs, which triggered the Electronic Privacy Information Centre for the Federal Trade Commission to open an investigation into Google's Cloud Computing Services [18]. Another example of a risk to data integrity recently occurred in Amazon S3 where users suffered from data corruption [62]. Further examples giving details of attacks can be read in [18],[52],[62].

Cachin et al. [18] argue that when multiple clients use cloud storage or when multiple devices are synchronized by one user, it is difficult to address the data corruption issues. One of the solutions that is proposed [18] is to use a Byzantine fault-tolerant replication protocol within the cloud. Hendricks et al. [31] state that this solution can avoid data corruption caused by some components in the cloud. However, Cachin et al. [18] claim that using the Byzantine fault-tolerant replication protocol within the cloud is unsuitable due to the fact that the servers belonging to cloud providers use the same system installations and are physically located in the same place.

Although this protocol solves the problem from a cloud storage perspective, Cachin et al. [18] argue that they remain concerned about the users' view, due to the fact that users trust the cloud as a single reliable domain or as a private cloud without being aware of the protection protocols used in the cloud provider's servers. As a solution, Cachin et al. [18] suggest that using Byzantine fault-tolerant protocols across multiple clouds from different providers is a beneficial solution.

B. *Data Confidentiality*

According to Garfinkel [26], another security risk that may occur with a cloud provider, such as the Amazon cloud service, is a hacked password or data confidentiality. If someone gains access to an Amazon account password, they will be able to access all of the account's instances and resources. Thus, the stolen password allows the hacker to erase all the information inside any virtual machine instances for the stolen user account, modify it, or even disable its services. Furthermore, there is a possibility for the user's email (Amazon user name) to be hacked (see [25] for a discussion of the potential risks of email), and since Amazon allows a lost password to be reset by email, the hacker may still be able to log in to the account after receiving the new reset password.

C. *Service Availability*

Another major concern in cloud services is service availability. Amazon [10] mentions in its licensing agreement that it is possible that the service might be unavailable from time to time. The user's web service may terminate for any reason at any time if any user's files break the cloud storage policy. In addition, if any damage occurs to any Amazon web service and the service fails, there will be no charge to the Amazon Company for this failure. Companies seeking to protect services from such failure need measures such as backups or use of multiple providers [26]. Both Google Mail and Hotmail experienced service down-time recently [18]. If a delay affects payments from users for cloud storage, the users may not be able to access their data. Due to a system administrator error, 45% of stored client data was lost in LinkUp (MediaMax) as a cloud storage provider [18].

Garfinkel [26] argues that information privacy is not guaranteed in Amazon S3. Data authentication which assures that the returned data is the same as the stored data is extremely important. Garfinkel [26] claims that instead of following Amazon's advice that organizations encrypt data before storing them in Amazon S3, organizations should use HMAC [38] technology or a digital signature to ensure data is not modified by Amazon S3. These technologies protect users from Amazon data modification and from hackers who may have obtained access to their email or stolen their password [26].

IV. MULTI-CLOUDS COMPUTING

This section will discuss the migration of cloud computing from single to multi-clouds to ensure the security of the user's data.

A. *Multi-Clouds: Preliminary*

The migration of cloud computing from single toward multi-clouds to ensure the security of user's data is extremely important [6]. The term "multi-clouds" is similar to the terms "interclouds" or "cloud-of-clouds" that were introduced by [66]. These terms suggest that cloud computing should not end with a single cloud. Using their illustration, a cloudy sky incorporates different colors and shapes of clouds which leads to different implementations and administrative domains.

Recent research has focused on the multi-clouds environment [3], [13], [16], [17] which control several clouds and avoids dependency on any one individual cloud.

Cachin et al. [17] identify two layers in the multi-clouds environment: the bottom layer is the inner-cloud, while the second layer is the inter-cloud. In the inter-cloud, the Byzantine fault tolerance finds its place. We will first summarize the previous Byzantine protocols over the last three decades.

B. *Introduction of Byzantine Protocols*

In cloud computing, any faults in software or hardware are known as Byzantine faults that usually relate to inappropriate behavior and intrusion tolerance. In addition, it also includes arbitrary and crash faults [66]. Much research has been dedicated to Byzantine fault tolerance (BFT) since its first introduction [40], [49]. Although BFT research has received a great deal of attention, it still suffers from the limitations of practical adoption [39] and remains peripheral in distributed systems [66].

The relationship between BFT and cloud computing has been investigated, and many argue that in the last few years, it has been considered one of the major roles of the distributed system agenda. Furthermore, many describe BFT as being of only "purely academic interest" for a cloud service [14]. This lack of interest in BFT is quite different to the level of interest shown in the mechanisms for tolerating crash faults that are used in large-scale systems. Reasons that reduce the adoption of BFT are, for example, difficulties in design, implementation, or understanding of BFT protocols [66].

As mentioned earlier, BFT protocols are not suitable for single clouds. [66] argues that one of the limitations of BFT for the inner-cloud is that BFT requires a high level of failure independence, as do all fault-tolerant protocols [57]. If Byzantine failure occurs to a particular node in the cloud, it is reasonable to have a different operating system, different implementation, and different hardware to ensure such failure does not spread to other nodes in the same cloud. In addition, if an attack happens to a particular cloud, this may allow the attacker to hijack the particular inner-cloud infrastructure [66].

C. DepSky System: Multi-Clouds Model

This section will explain the recent work that has been done in the area of multi-clouds. Bessani et al. [13] present a virtual storage cloud system called DepSky which consists of a combination of different clouds to build a cloud-of-clouds. The DepSky system addresses the availability and the confidentiality of data in their storage system by using multi-clouds providers, combining Byzantine quorum system protocols, cryptographic secret sharing and erasure codes [13].

- *DepSky Architecture*

The DepSky architecture [13] consists of four clouds and each cloud uses its own particular interface. The DepSky algorithm exists in the clients' machines as a software library to communicate with each cloud (Figure 2). These four clouds are storage clouds, so there are no codes to be executed. The DepSky library permits reading and writing operations with the storage clouds.

DepSky Data model: As the DepSky system deals with different cloud providers, the DepSky library deals with different cloud interface providers and consequently, the data format is accepted by each cloud. The DepSky data model consists of three abstraction levels: the conceptual data unit, a generic data unit, and the data unit implementation. The reason why writers can fail arbitrarily is that even if their protocols tolerated writes in clouds, the faulty writers in their data unit, which lead to the corruption of the application that used the DepSKy system, will write wrong values [13].

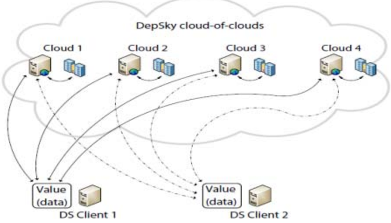

Figure 2:DepSky Architecture [13].

*DepSKy System model***:** The DepSky system model contains three parts: readers, writers, and four cloud storage providers, where readers and writers are the client's tasks. Bessani et al. [13] explain the difference between readers and writers for cloud storage. Readers can fail arbitrarily (for example, they can fail by crashing, they can fail from time to time and then display any behavior) whereas, writers only fail by crashing.

*Cloud storage providers in DepSky system***:** Passive cloud storage that does not execute any code in it supports five operations such as, "list (lists the files of a container in the cloud), get (reads a file), create (creates a container), put (writes or modifies a file in a container) and remove (deletes a file)" [13]. Access control in DepSky supplied by the system to guarantee that readers are those permitted to ask for the list and get operations to avoid data corruption, data intrusion, or unauthorized access, which lead to fail in Byzantine way [40]. The Byzantine protocols involve a set of storage clouds (n) where n = 3 f +1, and f is maximum number of clouds which could be faulty. In addition, any subset of (n – f) storage cloud creates byzantine quorum protocols [13].

- *Protocol Design foundation:*

Most protocols which provide the spine for the cloud storage system are Byzantine quorum protocols [21]. Bessani et al. [13] presented different kinds of protocols that implement Byzantine fault-tolerant (BFT) storage but they require code execution in their servers, which is not available in storage clouds [19],[28],[31],[43],[44], whereas in the DepSky protocols, the data and the metadata are written in different quorums. To overcome the limitations of the lack of server code able to verify the version number of the data being written, Bessani et al. [13] implemented a single-writer multi-reader register, which is suitable for many applications. On the other hand, Bessani et al. [13] showed different kinds of quorum protocols that do not consider the individual storage clouds as servers, however, they consider them as disks or passive shared memory objects [2], [12], [22] and [32]. These protocols need many steps to access the shared memory, which makes it difficult to communicate with the DepSky system as geographically discrete distributed systems due to the highly changeable latencies concerned [13]. To read or write the metadata and the data files in the data unit, DepSky protocols require two communication round-trips to implement them.

Bessani et al. [13] used a secret sharing scheme [59] to assure confidentiality of the stored data on the cloud without the need for a key distribution service. This schema aims to divide the secret into n shares where each share contains the secret. In addition, to ensure the secret, the n shares should be n >= f +1 shares (in this case f=1 faulty cloud). To recover the secret, knowledge of the different shares of the secret is needed, whereas with f number of shares or less, no information about the secret will be explored.

The secret sharing algorithm and the replication protocol are integrated where just a written data in the shares, in addition to the metadata, will be delivered to each storage cloud. The benefit of this is to assure there will be no access to the stored data in the cloud from the cloud provider except for the authorized client, which can access the shares of at least f+1 different clouds and can reconstruct the original data that are hidden in the shares [13].

Bessani et al. [13] assume that using secret sharing schema permitted them to guarantee confidentiality along with the stored data without using the mechanism of key distribution which allows sharing of the secret key between readers and writers of a data unit. In addition, the DepSky mechanism is able to choose which readers can access the data unit by the cloud provider's access control mechanism [13].

The result of replicating and storing the data on n clouds in DepSky would be costly, however, [13] combine the secret sharing schema along with the erasure code algorithm to reduce the size of the original data for each share by a factor of n/f+1 [51]. The original proposal of [37] was followed using this combination to have an information-efficient secret sharing schema, where a random secret key encrypts the data, and then this encrypted data is encoded. Consequently, the key is divided by using a secret sharing schema and is then distributed into the server and receives a share of the key and a block of the encrypted data.

Depsky implemented two protocols for his system [13]: the first is called DepSky-A that replicates the data into different cloud providers by using quorum techniques to improve the availability and the integrity of the stored data in the cloud. However, there are two limitations in this protocol: first, replicating data into a multi-cloud will have a high cost, more than if it was stored in a single cloud; and secondly, this protocol lacks data confidentiality as the data is stored in the storage cloud without any change of its values. To address these limitations, [13] proposed the second protocol called DepSky-CA. DepSky-CA ensures data confidentiality by using an information-efficient secret sharing scheme [37] that combines a secret sharing scheme and an optimal erasure code. This schema was used to divide the data into a set of blocks where firstly, f+1 blocks were enough to obtain the original data, secondly, f or less blocks were not enough to know the original stored data [13].

DepSky-A and DepSky-CA protocols differ in several ways. Firstly, in DepSky CA data encryption prevents the individual cloud from obtaining the stored data. Secondly, storing the encryption key in the cloud by using a secret sharing schema prevents the faulty cloud from reconstructing it. Thirdly, reducing the size of the stored data for DepSky-CA to halve the cost of using DepSky-A in each cloud due to the use of the erasure code scheme that only has access to f+1 clouds in the best scenario, means it can only store half the total amount of data in each cloud. In other words, DepSky-A stores 4 times (of its capacity) and DepSky-CA stores double times (of its capacity) and a single cloud stores a single copy due to the storage costs associated with the amount of data stored in the cloud [13].

D. Analysis of Multi-Clouds Research

Moving from single clouds or inner-clouds to multi-clouds is reasonable and important for many reasons. According to Cachin et al. [18], "Services of single clouds are still subject to outage". In addition, [16] showed that over 80% of company management "fear security threats and loss of control of data and systems". Vukolic [66] assumes that the main purpose of moving to interclouds is to improve what was offered in single clouds by distributing reliability, trust, and security among multiple cloud providers. In addition, reliable distributed storage [21] which utilizes a subset of BFT techniques was suggested by Vukolic [66] to be used in multi-clouds. A number of recent studies in this area have built protocols for interclouds. RACS (Redundant Array of Cloud Storage) [3] for instance, utilizes RAID-like techniques that are normally used by disks and file systems, but for multiple cloud storage. Abu-Libdeh et al. [3] assume that to avoid "vender lock-in", distributing a user's data among multiple clouds is a helpful solution. This replication also decreases the cost of switching providers and offers better fault tolerance. Therefore, the storage load will be spread among several providers as a result of the RACS proxy [3].

HAIL (High Availability and Integrity Layer) [16] is another example of a protocol that controls multiple clouds. HAIL is a distributed cryptographic system that permits a set of servers to ensure that the client's stored data is retrievable and integral. HAIL provides a software layer to address availability and integrity of the stored data in an intercloud [16].

Cachin et al. [17] present a design for intercloud storage (ICStore), which is a step closer than RACS and HAIL as a dependable service in multiple clouds. Cachin et al. [17] develop theories and protocols to address the CIRC attributes (confidentiality, integrity, reliability and consistency) of the data stored in clouds.

As mentioned before, Bessani et al. [13] present a virtual storage cloud system called DepSky consisting of a combination of different clouds to build a cloud-of-clouds. Bessani et al. [13] discuss some limitations of the HAIL protocol and RACS system when compared with DepSky. HAIL does not guarantee data confidentiality, it needs code execution in their servers, and it does not deal with multiple versions of data. None of these limitations are found in DepSky [13], whereas the RACS system differs from the DepSky system in that it deals with "economic failures" and vendor lock-in and does not address the issue of cloud storage security problems. In addition, it also does not provide any mechanism to ensure data confidentiality or to provide updates of the stored data. Finally, the DepSky system presents an experimental evaluation with several clouds, which is different from other previous work on multi-clouds [13].

There are a number of studies which is related to untrusted clouds. For instance, similar to DepSky, Depot improves the flexibility of cloud storage, as Mahajan et al. (2010) believe that cloud storages face many risks. However, Depot provides a solution that is cheaper due to using single clouds, but it does not tolerate losses of data and its service availability depends on cloud availability [13]. Other work which implements services on top of untrusted clouds are studies such as SPORC [24] and Venus [60]. These studies are different from the DepSky system because they consider a single cloud (not a cloud-of-clouds). In addition, they need code execution in their servers. Furthermore, they offer limited support for the unavailability of cloud services in contrast to DepSky [13].

E. Current Solutions for Security Risks

In order to reduce the risk in cloud storage, customers can use cryptographic methods to protect the stored data in the cloud [18]. Using a hash function [46] is a good solution for data integrity by keeping a short hash in local memory. In this way, authentication of the server responses is done by recalculating the hash of the received data which is compared with the local stored data [18]. If the amount of data is large, then a hash tree is the solution [46]. Many storage system prototypes have implemented hash tree functions, such as SiRiUS [27] and TDB [42]. Mykletun et al. [47] and Papamanthou et al. [48] claim that this is an active area in research on cryptographic methods for stored data authentication. Cachin et al. [18] argue that although the previous methods allow consumers to ensure the integrity of their data which has been returned by servers, they do not guarantee that the server will answer a query without knowing what that query is and whether the data is stored correctly in the server or not. Proofs of Retrievability (PORs) and Proofs of Data Possession (PDP) are protocols introduced by [34] and [11] to ensure high probability for the retrieval of the user's data. Cachin et al. [18] suggest using multiple cloud providers to ensure data integrity in cloud storage and running Byzantine-fault-tolerant protocols on them where each cloud maintains a single replica [20], [31]. Computing resources are required in this approach and not only storage in the cloud, such a service provided in Amazon EC2, whereas if only storage service is available, Cachin et al. [18] suggest working with Byzantine Quorum Systems [43] by using Byzantine Disk Paxos [2] and using at least four different clouds in order to ensure users' atomicity operations and to avoid the risk of one cloud failure.

As mentioned earlier, the loss of availability of service is considered one of the main limitations in cloud computing and it has been addressed by storing the data on several clouds. The loss of customer data has caused many problems for many users such as the problem that occurred in October 2009 when the contacts, photos, etc. of many users of the Sidekick service in Microsoft were lost for several days [56].

Bessani et al. [13] use Byzantine fault-tolerant replication to store data on several cloud servers, so if one of the cloud providers is damaged, they are still able to retrieve data correctly. Data encryption is considered the solution by [13] to address the problem of the loss of privacy. They argue that to protect the stored data from a malicious insider, users should encrypt data before it is stored in the cloud. As the data will be accessed by distributed applications, the DepSky system stores the cryptographic keys in the cloud by using the secret sharing algorithm to hide the value of the keys from a malicious insider.

In DepSky system, data is replicated in four commercial storage clouds (Amazon S3, Windows Azure, Nirvanix and Rackspace); it is not relayed on a single cloud, therefore, this avoids the problem of the dominant cloud causing the so-called vendor lock-in issue [3]. In addition, storing half the amount of data in each cloud in DepSky system is achieved by the use of erasure codes. Consequently, exchanging data between provider to another one will result in a smaller cost. DepSky system aims to reduce the cost of using four clouds (which is four times the overhead) to twice the cost of using a single cloud, which is a significant advantage [13].

F. Limitation of Current Solutions

The problem of the malicious insiders in the cloud infrastructure which is the base of cloud computing is considered by [54]. IaaS cloud providers provide the users with a set of virtual machines from which the users can benefit by running software on them. The traditional solution to ensure data confidentiality by data encryption is not sufficient due to the fact that the user's data needs to be manipulated in the virtual machines of cloud providers which cannot happen if the data has been encrypted [54]. Administrators manage the infrastructure and as they have remote access to servers, if the administrator is a malicious insider, then he can gain access to the user's data [41]. Van Dijk and Juels [64] present some negative aspects of data encryption in cloud computing. In addition, they assume that if the data is processed from different clients, data encryption cannot ensure privacy in the cloud.

Although cloud providers are aware of the malicious insiders danger, they assume that they have critical solutions to alleviate the problem [29]. Rocha and Correia [54] determine possible attackers for IaaS cloud providers. For example, Grosse et al. [29] propose one solution by preventing any physical access to the servers. However, Rocha and Correia [54] argue that the attackers outlined in their work have remote access and do not need any physical access to the servers. Grosse et al. [29] propose another solution by monitoring all access to the servers in a cloud where the user's data is stored. However, Rocha and Correia [54] claim that this mechanism is beneficial for monitoring employee's behavior in terms of whether they are following the privacy policy of the company or not, but it is not effective because it detects the problem after it has happened.

Rocha and Correia [54] classified four types of attacks that can affect the confidentiality of the user's data in the cloud. These four types of attacks could occur when the malignant insider can determine text passwords in the memory of a VM, cryptographic keys in the memory of VM files, and other confidential data. In addition, they argue that the recent research mechanisms are not good enough to consider the issue of data confidentiality and to protect data from these attacks. This does not mean that these mechanisms are not useful; rather that they do not focus on solving the problems that [54] address in their research. Some of the solutions [50] are used as part of cloud computing solutions, while different types of solutions focus on solving the whole data confidentiality issue intrinsic to cloud computing [13], [55]. Rocha and Correia [54] suggests trusted computing and distributing trust among several cloud providers as a novel solution to solving security problems and challenges in cloud computing. The idea of replicating data among different clouds has been applied in the single system DepSky [13]. Rocha and Correia [54] present the limitations of this work which occurs due to the fact that DepSky is only a storage service like Amazon S3, and does not offer the IaaS cloud model. On the other hand, this system provides a secure storage cloud, but does not provide security of data in the IaaS cloud model. This is because it uses data encryption and stores the encrypted key in the clouds by using a secret sharing technique, which is inappropriate for the IaaS cloud model [54].

Table 1 details the security risks addressed in the previous research and the mechanisms that have been proposed as a solution for these security risks in the cloud computing environment. Security risk issues in cloud computing have attracted much research interest in recent years.

It is clear from the table that in the past more research has been conducted into single clouds than into multi-clouds. Multi–clouds can address the security issues that relate to data integrity, data confidentiality, and service availability. In addition, most of the research has focused on providing secure "cloud storage" such as in DepSky. Therefore, providing a cloud database system like in our MCDB model [8], instead of normal cloud storage, is a significant goal in order to run queries and deal with databases; in other words, to profit from a database-as-a-service facility in a cloud computing-environment. The reliability system has improved in our MCDB model [7] by using Triple modular redundancy (TMR) technique [33].

Based on the references cited in table 1, we computed that in 2009, 67% of the research on security in cloud computing addressed the issue of a single cloud, whereas 33% of the research in the same year addressed the issue of multi-clouds. And in 2010, 80% of research focused on single clouds while only 20% or research was directed in the area of multi-clouds.

V. Future Work

For future work, we aim to provide a framework to supply a secure cloud database that will reduce the security risks facing the cloud computing community. This framework will apply multi-clouds and the secret sharing algorithm to reduce the risk of data intrusion and the loss of service availability in the cloud and it will also ensure data integrity.

In relation to data confidentiality and data integrity, suppose we want to distribute the data into three different cloud providers, and we apply the secret sharing algorithm on the stored data in the cloud provider. An intruder needs to retrieve at least three values to be able to find out the real value that we want to hide from the intruder. This depends on Shamir's secret sharing algorithm with a polynomial function technique which claims that even with full knowledge of (k – 1) clouds, the service provider will not have any knowledge of vs (vs is the secret value) [59]. We have used this technique in previous databases-as-a-service research [5]. In other words, hackers need to retrieve all the information from the cloud providers to know the real value of the data in the cloud. Therefore, if the attacker hacked one cloud provider's password or even two cloud provider's passwords, they still need to hack the third cloud provider (in the case where k = 3) to know the secret which is the worst case scenario. Hence, replicating data into multi-clouds by using a multi-share technique may reduce the risk of data intrusion and increase data integrity. In other words, it will decrease the risk of the Hyper-Visor being hacked and Byzantine fault-tolerant data being stolen from the cloud provider.

Regarding service availability risk or loss of data, if we replicate the data into different cloud providers, we could argue that the data loss risk will be reduced. If one cloud provider fails, we can still access our data live in other cloud providers. To answer the question on how we can ensure a backup even if instances are down, [26] suggested for running services on multiple instances in Amazon EC2 and storing data in multiple Amazon S3, then link different Amazon Web Services (AWS) to different email's addresses used as a user name. There will be a dilemma if, for example, Amazon decides to delete user's data for any reason from their all instances depend on their web service licensing agreement (WSLA)[9]. The use of multiple cloud service providers may reduce the risk of loss of data. This fact has been discovered from this survey and we will explore dealing with different cloud provider interfaces and the network traffic between cloud providers.

VI. Conclusion

It is clear that although the use of cloud computing has rapidly increased, cloud computing security is still considered the major issue in the cloud computing environment. Customers do not want to lose their private information as a result of malicious insiders in the cloud. In addition, the loss of service availability has caused many problems for a large number of customers recently. Furthermore, data intrusion leads to many problems for the users of cloud computing.

The purpose of this work is to survey the recent research on single clouds and multi-clouds to address the security risks and solutions. We have found that much research has been done to ensure the security of the single cloud and cloud storage whereas multi-clouds have received less attention in the area of security. We support the migration to multi-clouds due to its ability to decrease security risks that affect the cloud computing user.

References

[1] (NIST), http://www.nist.gov/itl/cloud/, Accessed in May-2011.

[2] I. Abraham, G. Chockler, I. Keidar and D. Malkhi, Byzantine disk paxos: optimal resilience with Byzantine shared memory, Distributed Computing, 18 (2006), pp. 387-408.

[3] H. Abu-Libdeh, L. Princehouse and H. Weatherspoon, RACS: a case for cloud storage diversity, Proceedings of the 1st ACM symposium on Cloud computing, ACM, 2010, pp. 229-240.

[4] S. Akioka and Y. Muraoka, HPC benchmarks on Amazon EC2, Proceedings of The 2010 24th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2010, pp. 1029-1034.

[5] M. A. AlZain and E. Pardede, Using Multi Shares for Ensuring Privacy in Database-as-a-Service, Proceedings of The 2011 44th Hawaii International Conference on System Sciences (HICSS), IEEE, Kauai, USA, 2011, pp. 1-9.

[6] M. A. AlZain, B. Soh and E. Pardede, MCDB: Using Multi-clouds to Ensure Security in Cloud Computing, Proceedings of The 2011 Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), IEEE, Sydney, Australia, 2011, pp. 784-791.

[7] M. A. AlZain, B. Soh and E. Pardede, A New Approach Using Redundancy Technique to Improve Security in Cloud Computing, Proceedings of The 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec12), IEEE, Kuala Lumpur, Malaysia, 2012, pp. 230-235.
[8] M. A. AlZain, B. Soh and E. Pardede, A new model to ensure security in cloud computing services, Journal of Service Science Research, 4 (2012), pp. 49-70.
[9] Amazon, Amazon Web Services. Web services licensing agreement, (2010).
[10] Amazon, Amazon Web Services. Web services licensing agreement, Accessed in May-2011 (2006).
[11] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson and D. Song, Provable data possession at untrusted stores, Proceedings of the 14th ACM conference on Computer and communications security, ACM, 2007, pp. 598-609.
[12] H. Attiya and A. Bar-Or, Sharing memory with semi-byzantine clients and faulty storage servers, Proceedings The 2003 22nd International Symposium on Reliable Distributed Systems, 2003, pp. 371-378.
[13] A. Bessani, M. Correia, B. Quaresma, F. André and P. Sousa, DepSky: dependable and secure storage in a cloud-of-clouds, Proceedings of the sixth conference on Computer systems, ACM, 2011, pp. 31-46.
[14] K. Birman, G. Chockler and R. van Renesse, Toward a cloud computing research agenda, SIGACT News, 40 (2009), pp. 68-80.
[15] M. P. Boss G, Quan D, Legregni L, Hall H. , Cloud computing, White Paper, IBM (2007).
[16] K. D. Bowers, A. Juels and A. Oprea, HAIL: A high-availability and integrity layer for cloud storage, Proceedings of the 16th ACM conference on Computer and communications security, ACM, 2009, pp. 187-198.
[17] C. Cachin, R. Haas and M. Vukolic, Dependable storage in the Intercloud, IBM Research, 3783 (2010), pp. 1-6.
[18] C. Cachin, I. Keidar and A. Shraer, Trusting the cloud, ACM SIGACT News, 40 (2009), pp. 81-86.
[19] C. Cachin and S. Tessaro, Optimal resilience for erasure-coded Byzantine distributed storage, Distributed Computing, 3724 (2005), pp. 497-498.
[20] M. Castro and B. Liskov, Practical Byzantine fault tolerance, Operating Systems Review, 33 (1998), pp. 173-186.
[21] G. Chockler, R. Guerraoui, I. Keidar and M. Vukolic, Reliable distributed storage, Computer, 42 (2009), pp. 60-67.
[22] G. Chockler and D. Malkhi, Active disk paxos with infinitely many processes, Proceedings of The 2002 twenty-first annual symposium on Principles of distributed computing, ACM, 2002, pp. 78-87.
[23] Clavister, Security in the cloud, Clavister White Paper (2008), pp. 1-6.
[24] A. J. Feldman, W. P. Zeller, M. J. Freedman and E. W. Felten, SPORC: Group collaboration using untrusted cloud resources, Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation - OSDI (2010), pp. 337-350.
[25] S. L. Garfinkel, Email-based identification and authentication: An alternative to PKI?, IEEE Security and Privacy, 1 (2003), pp. 20-26.
[26] S. L. Garfinkel, An evaluation of amazon's grid computing services: EC2, S3, and SQS, http://simson.net/clips/academic/2007.Harvard.S3.pdf, 2007, pp. 1-15.
[27] E. J. Goh, H. Shacham, N. Modadugu and D. Boneh, SiRiUS: Securing remote untrusted storage, Proceedings of the Tenth Network and Distributed System Security (NDSS) Symposium, 2003, pp. 131–145.
[28] G. R. Goodson, J. J. Wylie, G. R. Ganger and M. K. Reiter, Efficient Byzantine-tolerant erasure-coded storage, Proceedings of the International Conference on Dependable Systems and Networks, 2004, pp. 1-22.
[29] E. Grosse, J. Howie, J. Ransome, J. Reavis and S. Schmidt, Cloud computing roundtable, Security & Privacy, IEEE, 8 (2010), pp. 17-23.
[30] J. Heiser, What you need to know about cloud computing security and compliance, Gartner, Research, ID (2009).
[31] J. Hendricks, G. R. Ganger and M. K. Reiter, Low-overhead byzantine fault-tolerant storage, Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles, ACM, 2007, pp. 73-86.
[32] P. Jayanti, T. D. Chandra and S. Toueg, Fault-tolerant wait-free shared objects, Journal of the ACM (JACM), 45 (1998), pp. 451-500.
[33] B. W. Johnson, Design & analysis of fault tolerant digital systems, Addison-Wesley Longman Publishing Co., Inc., 1988.
[34] A. Juels and B. S. Kaliski Jr, PORs: Proofs of retrievability for large files, Proceedings of the 14th ACM conference on Computer and communications security, ACM, 2007, pp. 584-597.
[35] S. Kamara and K. Lauter, Cryptographic cloud storage, Financial Cryptography and Data Security, 6054 (2010), pp. 136-149.
[36] L. M. Kaufman, Data security in the world of cloud computing, IEEE Security & Privacy, 7 (2009), pp. 61-64.
[37] H. Krawczyk, Secret sharing made short, Proceedings of the 13th annual international cryptology conference on Advances in cryptology Springer, 1994, pp. 136-146.
[38] H. Krawczyk, M. Bellare and R. Canetti, HMAC: Keyed-hashing for message authentication, in R. Editor, ed., 1997, pp. 1-11.
[39] P. Kuznetsov and R. Rodrigues, BFTW 3: why? when? where? workshop on the theory and practice of byzantine fault tolerance, ACM SIGACT News, 40 (2009), pp. 82-86.
[40] L. Lamport, R. Shostak and M. Pease, The Byzantine generals problem, ACM Transactions on Programming Languages and Systems (TOPLAS), 4 (1982), pp. 382-401.
[41] P. A. Loscocco, S. D. Smalley, P. A. Muckelbauer, R. C. Taylor, S. J. Turner and J. F. Farrell, The inevitability of failure: The flawed assumption of security in modern computing environments, Proceedings of the 21st National Information Systems Security Conference, 1998, pp. 303-314.
[42] U. Maheshwari, R. Vingralek and W. Shapiro, How to build a trusted database system on untrusted storage, Proceedings of the 4th conference on Symposium on Operating System Design & Implementation, USENIX Association, 2000, pp. 10-10.
[43] D. Malkhi and M. Reiter, Byzantine quorum systems, Distributed Computing, 11 (1998), pp. 203-213.
[44] J. P. Martin, L. Alvisi and M. Dahlin, Minimal byzantine storage, Distributed Computing (2002), pp. 311-325.
[45] H. Mei, J. Dawei, L. Guoliang and Z. Yuan, Supporting Database Applications as a Service, Proceedings of the 2009 International Conference on Data Engineering IEEE 2009, pp. 832-843.
[46] R. C. Merkle, Protocols for public key cryptosystems, IEEE Symposium on Security and Privacy, IEEE, 1980, pp. 122-134.

[47] E. Mykletun, M. Narasimha and G. Tsudik, Authentication and integrity in outsourced databases, ACM Transactions on Storage (TOS), 2 (2006), pp. 107-138.
[48] C. Papamanthou, R. Tamassia and N. Triandopoulos, Authenticated hash tables, Proceedings of the 15th ACM conference on Computer and communications security, ACM, 2008, pp. 437-448.
[49] M. Pease, R. Shostak and L. Lamport, Reaching agreement in the presence of faults, Journal of the ACM (JACM), 27 (1980), pp. 228-234.
[50] R. Perez, R. Sailer and L. van Doorn, vTPM: virtualizing the trusted platform module, Proceedings of the 15th conference on USENIX Security Symposium 2006, pp. 305-320.
[51] M. O. Rabin, Efficient dispersal of information for security, load balancing, and fault tolerance, Journal of the ACM (JACM), 36 (1989), pp. 335-348.
[52] RedHat, https://rhn.redhat.com/errata/RHSA-2008-0855.html, Accessed in May-2011.
[53] T. Ristenpart, E. Tromer, H. Shacham and S. Savage, 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, ACM, 2009, pp. 199-212.
[54] F. Rocha and M. Correia, Lucy in the Sky without Diamonds: Stealing Confidential Data in the Cloud, Proceedings of The 2011 1st International Conference on Dependable Systems and Networks Workshops (DSN-W), IEEE, 2011, pp. 1-6.
[55] N. Santos, K. P. Gummadi and R. Rodrigues, Towards trusted cloud computing, Proceedings of the 2009 conference on Hot topics in cloud computing, USENIX Association, 2009, pp. 1-5.
[56] D. Sarno, Microsoft says lost sidekick data will be restored to users, Los Angeles Times, (2009).
[57] F. Schneider and L. Zhou, Implementing trustworthy services using replicated state machines, Replication (2010), pp. 151-167.
[58] H. A. Seccombe A, Meisel A, Windel A, Mohammed A , Licciardi A, etal., Security guidance for critical areas of focus in cloud computing, CloudSecurityAlliance, 2009, 25 p. (2009).
[59] A. Shamir, How to share a secret, Commun. ACM, 22 (1979), pp. 612-613.
[60] A. Shraer, C. Cachin, A. Cidon, I. Keidar, Y. Michalevsky and D. Shaket, Venus: Verification for untrusted cloud storage, Proceedings of the 2010 ACM workshop on Cloud computing security workshop, ACM, 2010, pp. 19-30.
[61] S. Subashini and V. Kavitha, A survey on security issues in service delivery models of cloud computing, Journal of Network and Computer Applications (2011), pp. 1-11.
[62] Sun, http://blogs.sun.com/gbrunett/entry/amazon_s3_silent_data_corruption, Accessed in May-2011.
[63] H. Takabi, J. B. D. Joshi and G. Ahn, Security and Privacy Challenges in Cloud Computing Environments, Security & Privacy, IEEE, 8 (2010), pp. 24-31.
[64] M. Van Dijk and A. Juels, On the impossibility of cryptography alone for privacy-preserving cloud computing, Proceedings of the 5th USENIX conference on Hot topics in security, 305 (2010), pp. 1-8.
[65] J. Viega, Cloud computing and the common man, Computer, 42 (2009), pp. 106-108.
[66] M. Vukolic The Byzantine empire in the intercloud, ACM SIGACT News, 41 (2010), pp. 105-111.
[67] C. Wang, Q. Wang, K. Ren and W. Lou, Ensuring data storage security in cloud computing, Proceedings of The 2009 17th International Workshop on Quality of Service, IEEE, 2009, pp. 1-9.
[68] B. Whyman, Cloud Computing, Information Security and Privacy Advisory Board (2008), pp. 11–13.

Mohammed A. AlZain is a PhD candidate in the Department of Computer Science and Computer Engineering at La Trobe University, Melbourne, Australia since Oct-2010. Currently, his PhD research is in Cloud Computing Security under Assoc/Prof. Ben Soh and Dr. Eric Pardede. He has achieved his Bachelor degree in Computer Science from King Abdulaziz University, Saudi Arabia in 2004, and then achieved his Master's degree in Information Technology from La Trobe University in 2010. He is a lecturer in the faculty of Computer Science and Information Technology at Al Taif University in Saudi Arabia. His areas of interest include but are not limited to: Cloud Computing Security, Distributed Systems, and Database as a service. His alternative email: Alzain50@hotmail.com.

Ben Soh is an Associate Professor in the Department of Computer Science and Computer Engineering at La Trobe University, Melbourne, Australia and a Senior Member of IEEE. He in 1995 obtained his PhD in Computer Science & Engineering at La Trobe. Since then, he has had numerous successful PhD graduates and published more than 150 peer-reviewed research papers. He has made significant contributions in various research areas, including fault-tolerant and secure computing, and web services.

Eric Pardede received the Master of Information Technology degree and Ph.D. degree in computer science from La Trobe University, Melbourne, Australia, in 2002 and 2006, respectively. He is currently a Lecturer with the Department of Computer Science and Computer Engineering, La Trobe University. He has wide range of teaching and research experience including in the area of databases, software engineering, information systems, entrepreneurship, and professional communication.

TABLE 1.
RELATED WORK ON CLOUD COMPUTING SECURITY

Ref	Year	Cloud Security	Addressed Security Risks			Privacy/ Security Mechanism	Type of cloud		Type of service	
			Data integrity	Data Confidentiality	Service availability		Single cloud	Multi clouds	Cloud storage	Cloud database
(AlZain,Pardede)	2011	√	√			Multi shares+ secret sharing algorithm				
(AlZain, et al)	2012	√	√	√	√	MCDB, (Multi shares+ secret sharing algorithm)		√		√
(AlZain, et al)	2012	√	√	√	√	MCDB, (Multi shares+ secret sharing algorithm+ TMR techniques)		√		√
(Bessani, et al.)	2011	√	√	√	√	DepSky,(Byzantine + secret sharing + cryptography)		√	√	
(Rocha,Correia)	2011	√ survey	√				√		√	
(Abu-Libdeh, et al.)	2010	√				RAID-like techniques+ introduced RACS		√	√	
(Cachin, et al.)	2010	√	√			ICStore ,(client-centric distributed protocols)		√	√	
(Feldman,et al.)	2010	√			√	SPORC, (fork)	√			
(Grosse, et al.)	2010	√								
(Kamara, Lauter)	2010	√				cryptography	√		√	
(Mahajan, et al.)	2010					Depot, (FJC)	√		√	
(A. Shraer, et al.)	2010	√	√			Venus	√		√	
(Subashini, Kavitha)	2010	√ survey	√		√		√		√	
(Takabi, et al.)	2010	√					√		√	
(Van Dijk, Juels)	2010	√	√				√		√	
(Bowers, et al.)	2009	√	√		√	HAIL (Proofs + cryptography)		√	√	
(Cachin, et al.)	2009	√ survey	√					√	√	
(Clavister)	2009	√	√			encrypted cloud VPN	√		√	
(Ristenpart, et al.)	2009	√					√		√	
(Santos, et al.)	2009	√	√		√	TCCP	√		√	
(Wang, et al.)	2009	√	√			homomorphic token + erasure-coded	√		√	
(Ateniese, et al.)	2007		√			PDP schemes				
(Garfinkel)	2007	√					√		√	