Journal of Computing and Information Technology - CIT 19, 2011, 1, 25–55
doi:10.2498/cit.1001864

A Review on Cloud Computing: Design Challenges in Architecture and Security

Fei Hu[1], Meikang Qiu[2], Jiayin Li[2], Travis Grant[1], Draw Tylor[1], Seth McCaleb[1], Lee Butler[1] and Richard Hamner [1]

[1] Department of Electrical and Computer Engineering, University of Alabama, Tuscaloosa, AL, USA
[2] Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA

Cloud computing is becoming a powerful network architecture to perform large-scale and complex computing. In this paper, we will comprehensively survey the concepts and architecture of cloud computing, as well as its security and privacy issues. We will compare different cloud models, trust/reputation models and privacy-preservation schemes. Their pros and cons are discussed for each cloud computing security and architecture strategy.

Keywords: cloud computing, security, privacy, computer networks, models

1. Introduction

Cloud computing is quickly becoming one of the most popular and trendy phrases being tossed around in today's technology world. "It's becoming the phrase du jour", says Gartner's Ben Pring [1]. It is the big new idea that will supposedly reshape the information technology (IT) services landscape. According to The Economist in a 2008 article, it will have huge impacts on the information technology industry, and also profoundly change the way people use computers [2]. What exactly is cloud computing then, and how will it have such a big impact on people and the companies they work for?

In order to define cloud computing, it is first necessary to explain what is referenced by the phrase "The Cloud". The first reference to "The Cloud" originated from the telephone industry in the early 1990s, when Virtual Private Network (VPN) service was first offered. Rather than hard-wire data circuits between the provider and customers, telephone companies began using VPN-based services to transmit data. This allowed providers to offer the same amount of bandwidth at a lower cost by rerouting network traffic in real-time to accommodate ever-changing network utilization. Thus, it was not possible to accurately predict which path data would take between the provider and customer. As a result, the service provider's network responsibilities were represented by a cloud symbol to symbolize the black box of sorts from the end-users' perspective. It is in this sense that the term "cloud" in the phrase cloud computing metaphorically refers to the Internet and its underlying infrastructure.

Cloud computing is in many ways a conglomerate of several different computing technologies and concepts like grid computing, virtualization, autonomic computing [40], Service-oriented Architecture (SOA) [43], peer-to-peer (P2P) computing [42], and ubiquitous computing [41]. As such, cloud computing has inherited many of these technologies' benefits and drawbacks. One of the main driving forces behind the development of cloud computing was to fully harness the already existing, but under-utilized computer resources in data centers. Cloud computing is, in a general sense, on-demand utility computing for anyone with access to the cloud. It offers a plethora of IT services ranging from software to storage to security, all available anytime, anywhere, and from any device connected to the cloud. More

formally, the National Institute of Standards and Technology (NIST) defines cloud computing as a model for convenient, on-demand network access to computing resources such as networks, servers, storage, applications, and services that can be quickly deployed and released with very little management by the cloud-provider [4]. Although the word "cloud" does refer to the Internet in a broad sense, in reality, clouds can be public, private, or hybrid (a combination of both public and private clouds). Public clouds provide IT services to anyone in the general public with an Internet connection and are owned and maintained by the company selling and distributing these services. In contrast, private clouds provide IT services through a privately-owned network to a limited number of people within a specific organization.

As two examples of public consumer-level clouds consider Yahoo!®Mail and YouTube. Through these two sites, users access data in the form of e-mails, attachments, and videos from any device that has an Internet connection. When users download e-mails or upload videos, they do not know where exactly the data came from or went. Instead, they simply know that their data is located somewhere inside the cloud. In reality, cloud computing involves much more complexity than the preceding two examples illustrate and it is this behind the scenes complexity that makes cloud computing so appealing and beneficial to individual consumers and large businesses alike.

Cloud computing is an emerging technology from which many different industries and individuals can greatly benefit. The concept is simple; the cloud (internet) can be utilized to provide services that would otherwise have to be installed on a personal computer. For example, service providers may sell a service to customers which would provide storage of customer information. Or perhaps a service could allow customers to access and use some software that would otherwise be very costly in terms of money and memory. Typically, cloud computing services are sold on an "as needed" basis, thus giving customers more control than ever before. Cloud computing services certainly have the potential to benefit both providers and users. However, in order for cloud computing to be practical and reliable, many existing issues must be resolved.

With ongoing advances in technology, the use of cloud computing is certainly on the rise. Cloud computing is a fairly recent technology that relies on the internet to deliver services to paying customers. Services that are most often used with cloud computing include data storage and accessing software applications (Figure 1). The use of cloud computing is particularly appealing to users for two reasons: it is rather inexpensive and it is very convenient. Users can access data or use applications with only a personal computer and internet access. Another convenient aspect of cloud computing is that software applications do not have to be installed on a user's computer, they can simply be accessed through the internet. However, as with anything else that seems too good to be true, there is one central concern with the use of cloud computing technology – security [44].

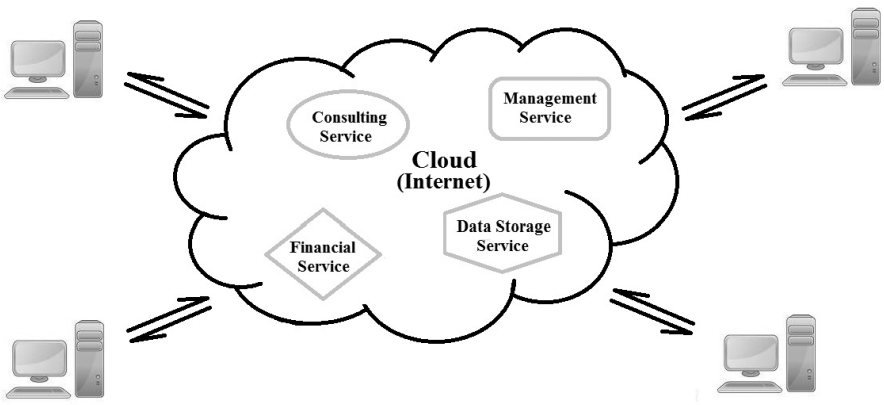

Figure 1. Cloud computing [5].

In the internet, people like to use email for communication because of its convenience, efficiency and reliability. It's well known that most of the contexts have no special meaning, which means it's more likely our daily communication. So, such context is less attractive for the hacker. However, when two people or two groups, even two countries, want to communicate using the internet secretly, for each communication partners they need some kind of encryption to ensure their privacy and confidentiality. For sake of that such situation occurs rarely, users don't want to make their processor slower by encrypting the email or other documents. There must be a way for the users to easily encrypt their document only when they require such service.

In the rest of the paper, we will first survey typical architectures from networks layers viewpoint. Then we will discuss cloud models. Next we move to security issues such as encryption-on-demand. The privacy preservation models will be described. In each part where multiple schemes are presented, we will also compare their pros and cons.

2. Cloud Computing Architecture

2.1. Abstract Layers

To begin understanding cloud computing in some sort of detail, it is necessary to examine it in abstraction layers beginning at the bottom and working upwards. Figure 2 illustrates the five layers that constitute cloud computing [44]. A particular layer is classified above another if that layer's services can be composed of services provided by the layer beneath it.

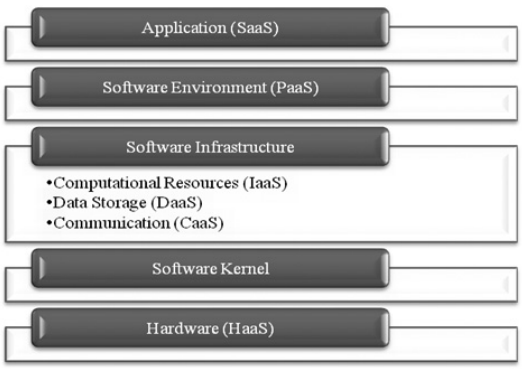

Figure 2. The five abstraction layers of cloud computing [5].

The bottom layer is the physical hardware, namely the cloud-provider owned servers and switches that serve as the cloud's backbone. Customers who use this layer of the cloud are usually big corporations who require an extremely large amount of subleased Hardware as a Service (HaaS). As a result, the cloud-provider runs, oversees, and upgrades its subleased hardware for its customers. Cloud-providers must overcome many issues related to the efficient, smooth, and quick allocation of HaaS to their customers, and one solution that allows providers to address some of these issues involves using remote scriptable boot-loaders. Remote scriptable boot-loaders allow the cloud-provider to specify the initial set of operations executed by the servers during the boot process, meaning that complete stacks of software can be quickly implemented by remotely located data center servers.

The next layer consists of the cloud's software kernel. This layer acts as a bridge between the data processing performed in the cloud's hardware layer and the software infrastructure layer which operates the hardware. It is the lowest level of abstraction implemented by the cloud's software and its main job is to manage the server's hardware resources while at the same time allowing other programs to run and utilize these same resources. Several different implementations of software kernels include operating system (OS) kernels, hypervisors, and clustering middleware. Hypervisors allow multiple OSs to be run on servers at the same time, while clustering middleware consists of software located on groups of servers, which allows multiple processes running on one or multiple servers to interact with one another as if all were concurrently operating on a single server.

The abstraction layer above the software kernel is called software infrastructure. This layer renders basic network resources to the two layers above it in order to facilitate new cloud software environments and applications that can be delivered to end-users in the form of IT services. The services offered in the software infrastructure layer can be separated into three different subcategories: computational resources, data storage, and communication. Computational resources, also called Infrastructure as a Service (IaaS), are available to

cloud customers in the form of virtual machines (VMs). Virtualization technologies such as para-virtualization [44], hardware-assisted virtualization, live-migration, and pause-resume enable a single, large server to act as multiple virtual servers, or VMs, in order to better utilize limited and costly hardware resources such as its central processing unit (CPU) and memory. Thus, each discrete server appears to cloud users as multiple virtual servers ready to execute the users' software stacks. Through virtualization, the cloud-provider who owns and maintains a large number of servers is able to benefit from gains in resource utilization and efficiency and claim the economies of scale that arise as a result. IaaS also automatically allocates network resources among the various virtual servers rapidly and transparently. The resource utilization benefits provided by virtualization would be almost entirely negated if not for the layer's ability to allocate servers' resources in real-time. This gives the cloud-provider the ability to dynamically redistribute processing power in order to supply users with the amount of computing resources they require without making any changes to the physical infrastructure of their data centers. Several current examples of clouds that offer flexible amounts of computational resources to its customers include the Amazon Elastic Compute Cloud (EC2) [9], Enomaly's Elastic Computing Platform (ECP) [10], and RESERVOIR architecture [6]. Data storage, which is also referred to as Data-Storage as a Service (DaaS), allows users of a cloud to store their data on servers located in remote locations and have instant access to their information from any site that has an Internet connection. This technology allows software platforms and applications to extend beyond the physical servers on which they reside. Data storage is evaluated based on standards related to categories like performance, scalability, reliability, and accessibility. These standards are not all able to be achieved at the same time and cloud-providers must choose which design criteria they will focus on when creating their data storage system. RESERVOIR architecture allows providers of cloud infrastructure to dynamically partner with each other to create a seemingly infinite pool of IT resources while fully preserving the autonomy of technological and business management decisions [6].In the RESERVOIR architecture, each infrastructure provider is an autonomous business with its own business goals. A provider federates with other providers (i.e., other RESERVOIR sites) based on its own local preferences. The IT management at a specific RESERVOIR site is fully autonomous and governed by policies that are aligned with the site's business goals. To optimize this alignment, once initially provisioned, resources composing a service may be moved to other RESERVOIR sites based on economical, performance, or availability considerations. Our research addresses those issues and seeks to minimize the barriers to delivering services as utilities with guaranteed levels of service and proper risk mitigation.

Two examples of data storage systems are Amazon Simple Storage Service (Amazon S3) [11] and Zecter ZumoDrive [12]. The communication subcategory of the software infrastructure layer, dubbed Communication as a Service (CaaS), provides communication that is reliable, schedulable, configurable, and (if necessary) encrypted. This communication enables CaaS to perform services like network security, real-time adjustment of virtual overlays to provide better networking bandwidth or traffic flow, and network monitoring. Through network monitoring, cloud-providers can track the portion of network resources being used by each customer. This "pay for what you use" concept is analogous to traditional public utilities like water and electricity provided by utility companies and is referred to as utility computing. Voice over Internet Protocol (VoIP) telephones, instant messaging, and audio and video conferencing are all possible services which could be offered by CaaS in the future. As a result of all three software infrastructure subcomponents, cloud-customers can rent virtual server time (and thus their storage space and processing power) to host web and online gaming servers, to store data, or to provide any other service that the customer desires.

There are several Virtual Infrastructure Managements in IaaS, such as CLEVER [25], OpenQRM [8], OpenNebula [16], and Nimbus [19]. CLEVER aims to provide Virtual infrastructure Management services and suitable interfaces at the High-level Management layer to enable the integration of high-level features such as Public Cloud Interfaces, Contextualization, Security and Dynamic Resources provisioning. OpenQRM is an open-source platform for enabling

flexible management of computing infrastructures. It is able to implement a cloud with several features that allows the automatic deployment of services. It supports different virtualization technologies and format conversion during migration. OpenNebula is an open and flexible tool that fits into existing data center environments to build a Cloud computing environment. OpenNebula can be primarily used as a virtualization tool to manage virtual infrastructures in the data-center or cluster, which is usually referred as Private Cloud. Only the more recent versions of OpenNebula are trying to support Hybrid Cloud to combine local infrastructure with public cloud-based infrastructure, enabling highly scalable hosting environments. OpenNebula also supports Public Clouds by providing Cloud interfaces to expose its functionalities for virtual machine, storage and network management. Nimbus provides two levels of guarantees: 1) quality of life: users get exactly the (software) environment they need, and 2) quality of service: provision and guarantee all the resources the workspace needs to function correctly (CPU, memory, disk, bandwidth, availability), allowing for dynamic renegotiation to reflect changing requirements and conditions. In addition, Nimbus can also provision a virtual cluster for Grid applications (e.g. a batch scheduler, or a workflow system), which is also dynamically configurable, a growing trend in Grid Computing.

The next layer is called the software environment, or platform, layer and for this reason is commonly referred to as Platform as a Service (PaaS). The primary users of this abstract layer are the cloud application developers who use it as a means to implement and distribute their programs via the cloud. Typically, cloud application developers are provided with a programming-language-level environment and a pre-defined set of application programming interfaces (APIs) to allow their software to properly interact with the software environment contained in the cloud. Two such examples of current software environments available to cloud software developers are the Google App Engine and Salesforce.com's Apex code. Google App Engine provides both Java and Python runtime environments as well as several API libraries to cloud application developers [13], while Salesforce.com's Apex code is an object-oriented programming language that allows developers to run and customize programs that interact with Force.com platform servers [14]. When developers design their cloud software for a specific cloud environment, their applications are able to utilize dynamic scaling and load balancing as well as easily have access to other services provided by the cloud software environment provider like authentication and e-mail. This makes designing a cloud application a much easier, faster, and more manageable task. Another example of the PaaS is the Azure from Microsoft [46]. Applications for Microsoft's Azure are written using the .NET libraries, and compiled to the Common Language Runtime, a language-independent managed environment. The framework is significantly more flexible than AppEngine's, but still constrains the user's choice of storage model and application structure. Thus, Azure is intermediate between application frameworks like AppEngine and hardware virtual machines like EC2 [47]. The top layer of cloud computing above the software environment is the application layer. Dubbed as Software as a Service (SaaS), this layer acts as an interface between cloud applications and end-users to offer them on-demand and many times fee-based access to web-based software through their web browsers. Because cloud users run programs by utilizing the computational power of the cloud-provider's servers, the hardware requirements of cloud users' machines can often be significantly reduced. Cloud-providers are able to seamlessly update their cloud applications without requiring users to install any sort of upgrade or patch since all cloud software resides on servers located in the provider's data centers. Google Apps and ZOHO®are two examples of offerings currently available. In many cases the software layer of cloud computing completely eliminates the need for users to install and run software on their own computers. This, in turn, moves the support and maintenance of applications from the end-user to the company or organization that owns and maintains the servers that distribute the software to customers.

There are several other layer architectures in cloud computing. For example, Sotomayor et al. proposed a three-layer model [37]: cloud management, Virtual infrastructure (VI) management, and VM manager. Cloud management provides remote and secure interfaces for creating, controlling, and monitoring virtualized re-

sources on an infrastructure-as-a-service cloud. VI management provides primitives to schedule and manage VMs across multiple physical hosts. VM managers provide simple primitives (start, stop, suspend) to manage VMs on a single host. Wang et al. presented a cloud computing architecture, the Cumulus architecture [45]. In this architecture, there are also three layers: virtual network domain, physical network domain, and Oracle file system.

2.2. Management Strategies for Multiple Clouds [17]

In order to mitigate the risks associated with unreliable cloud services, businesses can choose to use multiple clouds in order to achieve adequate fault-tolerance – a system's ability to continue functioning properly in the event that single or multiple failures occur in some of its components. The multiple clouds utilized can either be a combination of 1) all public clouds or 2) public and private clouds that together form a single hybrid cloud. A lack of well-defined cloud computing standards means that businesses wishing to utilize multiple clouds must manage multiple, often unique cloud interfaces to achieve their desired level of fault-tolerance and computational power. This can be achieved through the formulation of cloud interface standards so that any public cloud available to customers utilizes at least one well-defined standardized interface. As long as all cloud-providers use a standardized interface, the actual technology and infrastructure implemented by the cloud is not relevant to the end-user. Also, some type of resource management architecture is needed to monitor, maximize, and distribute the computational resources available from each cloud to the business.

There are several multiple clouds managements, including the Intercloud Protocols [20], and some method of cloud federation [5, 7, 48]. The Intercloud Protocols consist of six layers: actual physical layer, physical metaphor layer, platform metaphor layer, communication layer, management layer and endpoints layer. The possibility of a worldwide federation of Clouds has been studied recently [3, 5, 48]. Some of the challenges ahead for the Clouds, like monitoring, storage, QoS, federation of different organizations, etc. have been previously addressed by grids. Clouds present, however, specific elements that call for standardization too; e.g. virtual images format or instantiation/migration APIs [5]. So enhancing existing standards is granted to ensure the required interoperability.

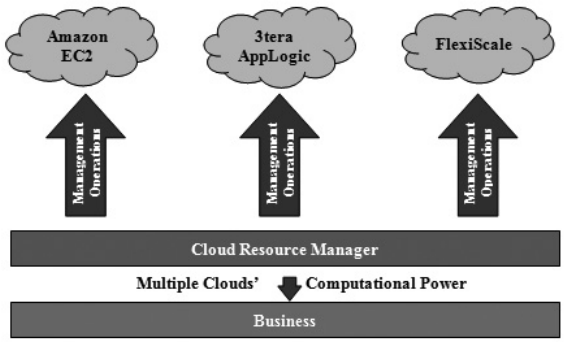

Figure 3. Architecture for resource management that allows fault-tolerance cloud computing.

Resource management architecture is realized by using abstract computational resources as its most basic building block. Next, a set of operations are defined so that each computational resource can be identified, started, stopped, or queried by the resource manager as to their processing state. Finally, the resource manager should provide some sort of interface to the IT employees of the company utilizing it to allow them to view, as well as manually modify, the amount of cloud computational resources being used by the company. Upon proper set-up and configuration of IT services from multiple cloud-providers, the resource manager should map the actual resources of each connected cloud to a generic, abstract representation of its computational resources. Ideally, once the actual resources of a cloud from a particular cloud-provider have been mapped, then any other cloud from that same cloud-provider could also be ported over to the resource manager as a generic computational resource representation using the same mapping scheme. Using the newly created generic representations of computational resources, the resource manager's algorithm could systematically identify each available resource, query its amount of utilized processing power, and then start or stop the resource accordingly in such a way as to maximize the total amount of cloud resources available to the business for a targeted price level. This process is depicted in Figure 3. Depending on each business's scenario and individual

preferences, the development of a suitable algorithm for the cloud manager to implement could be a seemingly complex task.

One such instance of a hybrid cloud resource manager has been realized through the collaboration of Dodda, Moorsel, and Smith [15]. Their cloud management architecture managed in tandem the Amazon EC2 via Query and Simple Object Access Protocol (SOAP) interfaces as well as their own private cloud via a Representational State Transfer (REST) interface. The Query interface of EC2 uses a query string placed in the Uniform Resource Locator (URL) to implement the management operations of the resource manager. Amazon, as the cloud-provider, provides a list of defined parameters and their corresponding values to be included in a query string by the resource manager. These query strings are sent by the cloud resource manager to a URL called out by the cloud-provider via HTTP GET messages in order to perform necessary management operations. In this way, EC2's interface is mapped to a generic interface that can be manipulated by the resource manager. The SOAP interface of EC2 operates very similarly to the Query interface. The same operations are available to resource managers using the SOAP interfaces as are available when using the Query interface, and HTTP GET messages are sent to a URL specified by the cloud-provider in order to perform cloud management operations. The only difference between the two interfaces is the actual parameters themselves that are needed to perform each operation.

The REST interface of the private cloud assigns a global identifier, in this case a Uniform Resource Identifier (URI), to each local resource. Each local resource is then manipulated via HTTP and mapped to its generic interface in much the same way as EC2's SOAP and Query interfaces. After each cloud-specific interface had been successfully mapped to its generic interface, the resource manager was able to effectively oversee both the EC2 and the private cloud at the same time by using a single, common set of interface commands.

However, one might wonder whether the choice of interface that the resource manager uses to oversee a cloud has a significant influence on the cloud's performance. To answer this question, a series of 1,000 identical management operation commands were issued by the resource manager to EC2 via the Query interface. Each command was issued a timestamp upon transmission from and arrival to the resource manager. Then, the same process was repeated for EC2 using the SOAP interface. A comparison of both interfaces' response times yielded a mean response time of 508 milliseconds (ms) for the Query interface and 905 ms for the SOAP interface. Thus, the SOAP interface's response time was nearly double the response time of the Query interface. In addition, the variance of the SOAP interface response times was much larger than the variance exhibited by the response times of the Query interface. So, the Query interface's response time is not only faster on average, but also more consistent than the response time of the SOAP interface. Based on these results, it should come as no surprise that most businesses use EC2's Query interface instead of its SOAP interface. This comparison clearly shows that when using a particular cloud, businesses must be careful in choosing the most responsive interface to connect to their resource manager so as to maximize performance and quality of service (QoS). The less amount of time required for processing commands issued by the resource manager, the more responsive the resource manager can be at maximizing and distributing the computational resources available from each cloud to the business.

Another example of hybrid cloud management described by H. Chen et al. utilizes a technique called intelligent workload factoring (IWF) [17]. As was the case with the previously described cloud resource manager approach, businesses can utilize IWF to satisfy highly dynamic computing needs. The architecture of the IWF management scheme allows the entire computational workload to be split into two separate types of loads, a base load and a trespassing load, when extreme spikes in the workload occur. The base load consists of the smaller, steady workload that is constantly demanded by users of cloud computing services while the trespassing load consists of the bigger, transient workload demanded on an unpredictable basis by users. Both loads are managed independently from one another in separate resource zones. The resources necessary to handle the base load can be obtained from a company's private cloud, and are thus referred to as the

base load resource zone, while the elastic computational power to meet the needs of the trespassing load can be accessed on-demand via a public cloud supplied by a cloud-provider, which is also known as the trespassing load resource zone. Base load management should proactively seek to use base zone resources efficiently by using predictive algorithms to anticipate the future base load conditions present at all times in the company's network. In contrast, trespass load management should have the ability to devote trespass zone resources to sudden, unpredictable changes in the trespass load in a systematic, yet agile fashion. Part of the IWF model's task is to ensure that the base load remains within the amount planned for by the company's IT department; in most cases, this is the computing capacity of the business's private cloud (i.e. on-site servers in data center). By doing this, the computational capacity that the business's on-site data center has to be overplanned for is considerably decreased, resulting in less capital expenditure and server underutilization for the business. The other part of the IWF model's job is to make sure a minimal amount of data replication is required to process the services obtained from the public cloud.

One instance of such an IWF management scheme was created by H. Chen and his colleagues [14]. Their system employed a private cloud in the form of a local cloud cluster as its base load resource zone and the Amazon EC2 as its trespassing load resource zone. The architecture used in this IWF hybrid cloud management system is depicted in Figure 4.

The IWF module, located at the very front of the management architecture, has a job that's twofold: first, it should send a smooth, constant workload to the base zone while at the same time avoid overloading either zone by excessively redirecting load traffic; second, the IWF module should effectively break down the overall load initiated by users of the network, both by type and amount of requested data. The process of load decomposition helps to avoid data redundancy in the management scheme. The base zone of the IWF system is on 100% of the time, and, as mentioned previously, the volume of the base load does not vary much with time. As a result, the resources of the private cloud that handle the base load are able to be run very efficiently. Nonetheless, a small amount of over-provisioning is still necessary in order for the local cloud cluster to be able to handle small fluctuations in the base load volume and to guarantee an acceptable QoS to the business' employees. The trespassing zone of the system is only on for X% of the time, where X is typically a small single-digit number. Because the

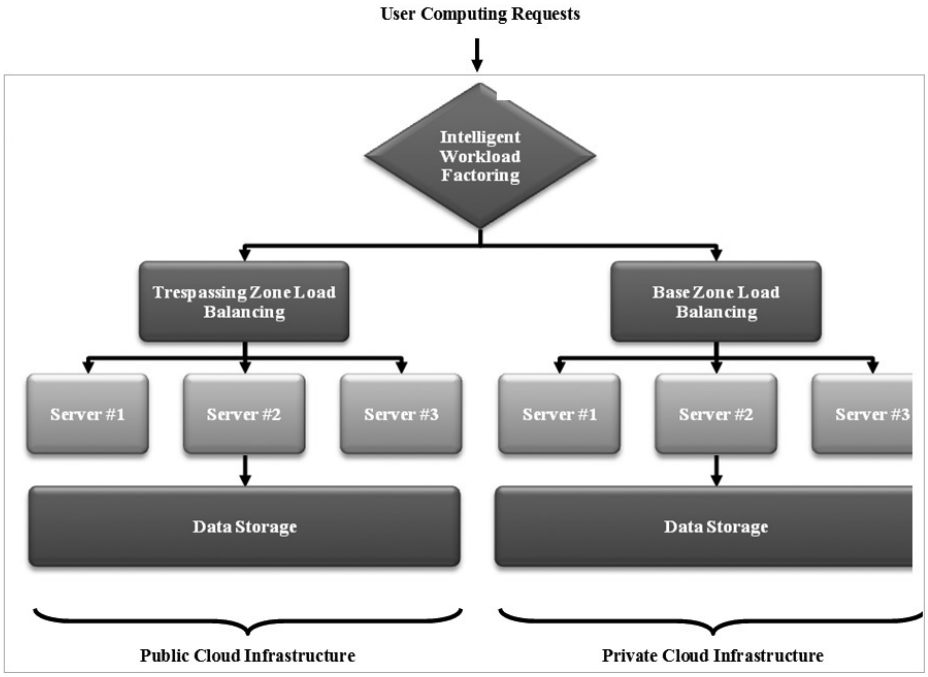

Figure 4. The IWF hybrid cloud management system [17].

trespassing zone must be able to adequately handle large spikes in processing demand, its computational capacity must be over-provisioned by a considerably large amount.

The procedure for implementing IWF can be modeled as a hypergraph partitioning problem consisting of vertices and nets. Data objects in the load traffic, such as streaming videos or tables in a database, are represented by vertices, while the requests to access these data objects are modeled as nets.

The link between a vertex and a net depicts the relationship that exists between a data object and its access request. Formally, a hypergraph is defined as

$$H = (V, N) \quad (1)$$

where V is the vertex set and N is the net set. Each vertex $v_i \in V$ is assigned weights w_i and s_i that represent the portion of the workload caused by the ith data object and the data size of the ith data object, respectively. The weight w_i is computed as the average workload per access request of the ith data object multiplied by its popularity. Each net $n_j \in N$ is assigned a cost c_j that represents the expected network overhead required to perform an access request of type j. The cost c_j attributed to each net is calculated as the expected number of type j access requests multiplied by the total data size of all neighboring data objects. The problem that must be solved in the design of the IWF management system involves assigning all data objects, or vertices, to two separate locations without causing either location to exceed its workload capacity while at the same time achieving a minimum partition cost. This is given by

$$Min(\sum_{n_j \in N_{\exp ected}} c_j + \gamma \sum_{v_i \in V_{trespas\sin g}} s_i) \quad (2)$$

where the first summation is the total cost of all nets that extend across multiple locations (i.e. the data consistency overhead), the second summation is the total data size of objects located in the trespassing zone (i.e. the data replication overhead), and γ is a weighting factor for each of the two summations.

The IWF management scheme consists of three main components: workload profiling, fast factoring, and base load threshold. A flow diagram of how these three components interact is shown in Figure 5. The workload profiling component continually updates the system load as changes occur to the number and nature of computing requests initiated by users of the network. By comparing the current system load with the base load threshold (the maximum load that the base zone can adequately handle), the workload profiling component places the system in one of two states – normal mode if the system load is less than or equal to the base load threshold or panic mode if the system load is greater than the base load threshold. The base load threshold can either be set manually by those in charge of the system or automatically by utilizing the past history of the base load. When the system is in normal mode, the fast factoring component forwards the current system load to the base zone. However, when the system is in panic mode, the fast factoring component implements a fast frequent data object detection algorithm to see if a computing request is seeking a data object that is in high demand.

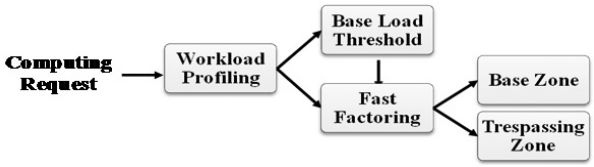

Figure 5. Flow diagram of IWF management scheme components [17].

If the object being requested is not in high demand, the fast factoring component forwards that portion of the system load to the base zone; if the requested object is in high demand, that portion of the system load is immediately forwarded to the trespassing zone. In essence, the fast factoring component removes the sporadic surges in computing power from the base zone (which is not equipped to handle such requests) and instead shifts them over to the trespassing zone (which is over provisioned to handle such spikes in the computing load) to be processed.

The fast frequent data object detection algorithm features a First In, First Out (FIFO) queue of the past c computing requests as well as two lists of the up-to-date k most popular data objects and historical k most popular data objects based on the frequency of access requests for each data object. The algorithm uses its queue and popular data object lists to accurately pinpoint data objects that are in high demand by the system The level of accuracy achieved by

the algorithm depends largely on how correct the counters are that measure the numbers of access requests for each data object For a data object T, its access request rate, $p(T)$, is defined as

$$p(T) = \frac{requests_T}{requests_{total}} \quad (3)$$

where $requests_T$ is the number of access requests for data object T and $requests_{total}$ is the total number of access requests for all data objects combined. The algorithm can approximate $p(\hat{T})$ (where the hat indicates an approximation instead of an exact result) such that

$$p(\hat{T}) \in \left(p(T) \cdot \left(1 - \frac{\beta}{2}\right), p(T) \cdot \left(1 + \frac{\beta}{2}\right)\right) \quad (4)$$

where β is a relatively small number that represents the percent error in the request rate approximation for a data object T. Obviously, a lower β (and thus percent error in approximating access request rates) allows the algorithm to more accurately determine which data objects are in high demand so that the frequent requests for such data objects can be passed on to the trespassing zone of the IWF management system and processed accordingly. Therefore, through the use of the fast frequent data object detection algorithm the performance of the IWF hybrid cloud management system is greatly improved.

As a way to compare the costs of different cloud computing configurations, H. Chen and his colleagues [17] used the hourly workload data from Yahoo! Video's web service measured over a 46-day period to serve as a sample workload. The workload (measured in terms of requested video streams per hour) contained a total of over 32 million stream requests. The three cloud computing configurations considered were: 1) a private cloud composed of a locally operated data center, 2) the Amazon EC2 public cloud, and 3) a hybrid cloud containing both a local cloud cluster to handle the bottom 95-percentile of the workload and the Amazon EC2 public cloud provisioned on-demand to handle the top 5-percentile of the workload. For the Amazon EC2, the price of a single machine hour was assumed to be $0.10. The cost results of the study are shown in Table 1, where for the sake of simplification, only the costs associated with running the servers themselves are considered.

From Table 1, implementing a private cloud that could handle Yahoo! Video's typical video streaming workload would require the yearly cost of maintaining 790 servers in a local data center. To handle the same workload using only a public cloud like Amazon EC2's would cost a staggering $1.3 million per year! Needless to say, although $0.10 per machine hour does not seem like a considerable cost, it adds up very quickly for a load of considerable size. Finally, the hybrid cloud configuration implementing IWF still maintains the ability to handle large fluctuations in load demand and requires a yearly cost of approximately $60,000 plus the cost of operating 99 servers in a local data center. This is by far the most attractive configuration among the three cloud computing models and should warrant strong consideration by any small or midsize business.

Cloud Computing Configuration	**Annual Cost**
Private Cloud	Cost of running a data center composed of 790 servers
Public Cloud	$1,384,000
Hybrid Cloud with IWF	$58,960 plus cost of running a data center composed of 99 servers

Table 1. Cost comparsion of three different cloud computing configurations [17].

2.3. Integrate Sensor Networks and Cloud Computing [18]

In addition to cloud computing, wireless sensor networks (WSNs) are quickly becoming a technological area of expansion. The ability of their autonomous sensors to collect, store, and send data pertaining to physical or environmental conditions while being remotely deployed in the field has made WSNs a very attractive solution for many problems in the areas of industrial monitoring, environmental monitoring, building automation, asset management, and health monitoring. WSNs have enabled a whole new wave of information to be tapped that has never before been available. By supplying this immense amount of real-time sensor data to people via software, the applications are limitless: up-to-date traffic information could be made available to commuters, the vital signs of soldiers on the battlefield could be monitored by medics, and environmental data pertaining to

earthquakes, hurricanes, etc. could be analyzed by scientists to help predict further disasters. All of the data made available using WSNs will need to be processed in some fashion. Cloud computing is a very real option available to satisfy the computing needs that arise from processing and analyzing all of the collected data.

A content-based publish/subscribe model for how WSNs and cloud computing can be integrated together is described in [18]. Publish/subscribe models do not require publishers, or sensor nodes, to send their messages to any specific receiver, or subscriber. Instead, sent messages are classified into classes and those who want to view messages, or sensor data, pertaining to a particular class simply subscribe to that class. By disconnecting the sensor nodes from its subscribers, the abilities of a WSN to scale in size and its nodes to dynamically change position are greatly increased. This is necessary as nodes are constantly reconfiguring themselves in an ad-hoc manner to react to suddenly inactive, damaged nodes as well as newly deployed and/or repositioned nodes. In order to deliver both real-time and archived sensor information to its subscribers, an algorithm that quickly and efficiently matches sensor data to its corresponding subscriptions is necessary. In addition, the computing power needed to process and deliver extremely large amounts of real-time, streaming sensor data to its subscribers may (depending on the transient spikes of the load) force cloud providers to be unable to maintain a certain QoS level. Therefore, a Virtual Organization (VO) of cloud-providers might be necessary to ensure that this scenario does not occur. In a VO, a dynamic partnership is formed with a certain understanding regarding the sharing of resources to accomplish individual but broadly related goals.

In the model described by M. M. Hassan et al. [18], sensor data is received by the gateway nodes of each individual WSN. These gateway nodes serve as the intermediary between the sensor nodes and a publisher/subscriber broker. The publisher/subscriber broker's job is to deliver sensor information to the SaaS cloud computing applications under dynamically changing conditions. In this way, sensor data from each WSN need only to be sent to one fixed location, the publisher/subscriber broker, instead of directly to multiple SaaS applications simultaneously; this reduces the amount of complexity necessary in the sensor and gateway nodes and allows for the cost of the nodes that comprise WSNs to remain minimal. The publisher/subscriber broker consists of four main components: the stream monitoring and processing component (SMPC), registry component (RC), analyzer component (AC), and disseminator component (DC). The SMPC receives the various streams of sensor data and initiates the proper method of analysis for each stream. The AC ascertains which SaaS applications the sensor data streams should be sent to and whether or not they should be delivered on a periodic or emergency basis. After proper determination, the AC passes this information on to the DC for delivery to its subscribers via SaaS applications. The DC component utilizes an algorithm designed to match each set of sensor data to its corresponding subscriptions so that it can ultimately distribute sensor data to users of SaaS applications. SaaS supplications are implemented using cloud resources and thus can be run from any machine that is connected to the cloud. SaaS packages the sensor data produced from WSNs into a usable graphical interface that can be examined and sorted by its users.

2.4. Typical Applications of Cloud Computing

In describing the benefits of cloud computing, one cannot help being bombarded by reasons related to economics. The two main reasons why cloud computing can be viewed as beneficial and worthwhile to individual consumers and companies alike are its cheap cost and efficient utilization of computing resources. Because cloud computing users only purchase the on-demand virtual server time and applications they need at little to no upfront cost, the capital expenditure associated with purchasing the physical network infrastructure is no longer required. Moreover, the physical space needed to house the servers and switches, the IT staff needed to operate, maintain, and upgrade the data center, and the energy costs of operating the servers are all suddenly deemed unnecessary.

For small businesses, barriers to enter markets that require significant amounts of computing power are substantially lowered. This means that small companies have newly found access to computing power that they otherwise would

never have been able to acquire. Because computing power costs are calculated based on usage, small businesses do not have to take on the unnecessary risk associated with committing a large amount of capital to purchasing network infrastructure. The ability of a company to grow or shrink the size of its workforce based on business demand without having to scale its network infrastructure capacity accordingly is another benefit of on-demand pricing and supply of computing power [21].

While operating completely without any on-location, physical servers might be an option for some small companies, many colleges and large companies might not consider this option very feasible. However, they could still stand to benefit from cloud computing by constructing a private cloud. Although a private cloud would still require on-site server maintenance and support and its relatively small size would prevent benefits related to economies of scale from being realized, the efficiency gains realized from server virtualization would be large enough in most cases to justify the effort. The IT research firm Infotech estimates that distributed physical servers "generally use only 20 percent of their capacity, and that, by virtualizing those server environments, enterprises can boost hardware utilization to between 60 percent and 80 percent" [22]. In addition, the private cloud could be connected to a public cloud as a source of on-demand computing power. This would help combat instances where spikes in computational power exceeded the capacity of the private cloud's virtualized servers. Part of section V, entitled 'Management Strategies for Multiple Clouds', explicates in considerable detail how multiple clouds, whether public or private, can be usefully utilized together in this manner.

2.5. Comparisons on Cloud Computing Models

From the preceding discussion it might seem as if cloud computing has virtually no downsides associated with it as long as the cloud is large enough in size to realize economies of scale; yet, cloud computing does have several drawbacks. Because cloud computing is still a new concept and still in its infant stages of development, there is currently a lack of well-defined market standards. This also means that a constant flow of new cloud-providers is entering the cloud computing market in an effort to each secure a large and loyal customer base.

As is the case with any developing technology, a constantly changing marketplace makes choosing the right cloud-provider and the proper cloud computing standards difficult. Customers face the risk of choosing a wrong provider who will soon go out of business and take the customer's data with them or backing a cloud computing standard that will eventually become obsolete. Once an individual consumer or business has chosen a particular provider and its associated cloud, it is currently difficult (but not altogether impossible) to transfer data between clouds if the customer wants to switch cloud-providers, and most likely, the customer will have to pay for the move in the form of switching costs.

Although virtualization has helped resource efficiency in servers, currently, performance interference effects still occur in virtual environments where the same CPU cache and translation lookaside buffer (TLB) hierarchy are shared by multiple VMs. The increasing number of servers possessing multi-core processors further compounds the effects of this issue. As a result of these interference effects, cloud-providers are currently not able to offer their customers an outright guarantee as to the specific level of computing performance that will be provided.

The physical location of a cloud-provider's servers determines the extent to which data stored on those servers is confidential. Due to recent laws such as the Patriot Act, files stored on servers that physically reside in the United States are subject to possible intense examination. For this reason, cloud-providers such as Amazon.com allow their customers to choose between using servers located in the United States or Europe.

Another negative aspect of cloud computing involves data transfer and its associated costs. Currently, because customers access a cloud using the Internet, which has limited bandwidth, cloud-providers recommend that users transfer large amounts of information by shipping their external storage devices to the cloud-provider. Upon receipt of the storage device, the cloud-provider uploads the customer's data

to the cloud's servers. For instance, Amazon S3 recommends that customers whose data will take a week or more to upload should use Amazon Web Services (AWS) Import/Export. This tool automates shipping an external storage device to AWS Import/Export using a mail courier such as United Parcel Service (UPS). For this service, the user is charged $80 per storage device handled and $2.49 per data-loading hour (where partial data-loading-hours are billed as full hours) [23]. Until the Internet infrastructure is improved several times over, transferring large amounts of data will be a significant obstacle that prevents some companies from adopting cloud computing.

There are several different cloud pricing models available, depending on the provider – the main three are tiered, per-unit, and subscription-based pricing. Amazon.com's clouds offer tiered pricing that corresponds to varying levels of offered computational resources and service level agreements (SLAs). SLAs are part of a cloud-provider's service contract and define specific levels of service that will be provided to its customers. Many cloud-providers utilize per-unit pricing for data transfers (as mentioned in the preceding paragraph's example of AWS Import/Export) and storage space. GoGrid Cloud Hosting, however, measures their server computational resources used in random access memory (RAM) per hour [24]. Subscription-based pricing models are typically used for SaaS. Rather than charge users for what they actually use, cloud-providers allow customers to know in advance what they will be charged so they can accurately predict future expenses.

Because transitioning between clouds is difficult and costly, end-users are somewhat locked in to whichever cloud-provider they choose. This presents reliability issues associated with the chosen cloud. Although rare and only for a matter of hours in most cases, some cloud access failures have already occurred with Google and Amazon.com's services. One such outage occurred with the Amazon EC2 as a result of a power failure at both its Virginia data center and its back-up data center in December 2009.

		Public Cloud	Private Cloud	Hybrid Cloud
Pros	1.	Simplest to implement and use	Allows for complete control of server software updates patches, etc.	Most cost-efficient through utilization flexibility of public and private clouds
	2.	Minimal upfront costs	Minimal long-term costs	Less susceptible to prolonged service outages
	3.	Utilization efficiency gains through server virtualization	Utilization efficiency gains through server virtualization	Utilization efficiency gains through server virtualization
	4.	Widespread accessibility	–	Suited for handling large spikes in workload
	5.	Requires no space dedicated for data center	–	–
	6.	Suited for handling large spikes in workload	–	–
Cons	1.	Most expensive long-term	Large upfront costs	Difficult to implement due to complex management schemes and assorted cloud center
	2.	Susceptible to prolonged services outages	Susceptible to prolonged services outages	Requires moderate amount of space dedicated for data center
	3.	–	Limited accessibility	–
	4.	–	Requires largest amount of space dedicated for data center	–
	5.	–	Not suited for handling large spikes in workload	–

Table 2. Summarization of pros and cons for public, private, and hybrid cloud.

This outage caused many east coast Amazon AWS users to be without service for approximately five hours. While these access issues are inevitable with any data center (including those on-site), lack of access to a business's cloud could occur at very inopportune instances resulting in a business's computing capability being effectively shut down for the duration of the service outage. This issue has the potential to nullify the hassle-free benefit that many cloud-providers tout as a selling point for utilizing cloud computing.

2.6. Summary of Cloud Computing Architecture

A common misconception among many of those who would consider themselves "tech-savvy" technology users is that cloud computing is a fancy word for grid computing; grid computing is a form of parallel computing where a group of networked computers form a single, large virtual supercomputer that can perform obscenely large amounts of computational operations that would take normal computers many years to complete. Although cloud computing is in some ways a descendant of grid computing, it is by no means an alternative title. Instead, cloud computing offers software, storage, and/or computing power as a service to its users, whether they be individuals or employees of a company. Due to its efficient utilization of computing resources and its ability to scale with workload demand, cloud computing is an enticing new technology to businesses of all sizes. However, the decision each business faces is not quite as clear-cut as whether cloud computing could prove beneficial to it. Instead, each business must make a decision as to what type of cloud it will utilize: public, private, or hybrid. Each cloud type has its own set of positives and negatives. These benefits and drawbacks are listed in Table 2. For example, the public cloud has the minimum upfront cost due to the fact that users do not need to invest in the infrastructure. However, it has the maximum long term cost since users have to pay for the lease in long term, while the user in the private cloud need to pay less in long term due to the ownership of the cloud.

In recent years, more and more companies have begun to adopt cloud computing as a way to cut costs without sacrificing their productivity. These cost savings stem from little or no upkeep required by their in-house IT departments as well as their ability to forego the huge capital expenditures associated with purchasing servers for their local data center. Even for those companies that adopt hybrid clouds which require some local servers, the number of these is dramatically reduced due to over provisioning of the data center no longer being necessary. In addition to cost savings, cloud computing allows companies to remain agile and more dynamically responsive to changes in forecasted business. If a company needs to quickly scale up or down the computing power of their network infrastructure, they can simply pay for more on-demand computing resources available from cloud-providers. This same level of scalability is simply not achievable with a local data center only. As cloud computing technology continues to mature, an increasing number of businesses are going to be willing to adopt it. For this reason, cloud computing is much more than just the latest technological buzzword on the scene – it's here to stay.

3. Cloud Computing Security

3.1. Why Security in Cloud Computing?

By using offloading data and cloud computing, a lot of companies can greatly reduce their IT cost. However, despite tons of merits of cloud computing, many companies owners began to worry about the security treats. Because in the cloud-based computing environment, the employees can easily access, falsify and divulge the data. Sometime such behavior is a disaster for a big and famous company.

Encryption is a kind of ideal way to solve such problem, whereas for the customers who are using the cloud computing system cannot use such encrypted data. The original data must be used in the host memory otherwise the host VM machine cannot do applications on-demand. For that sake, people can hardly achieve the good security in today's Cloud services. For example, the Amazon's EC2 is one of the service providers who have privileged to read and tamper the data from the customers. There is no security for the customers who use such service.

Some service providers develop some technical method aimed to avoid the security treats from the interior. For instance, some providers limit the authority to access and manage the hardware, monitor the procedures, and minimize the number of staff who has privilege to access the vital parts of the infrastructure. However, at the provider backend, the administrator can also access the customer's VM-machine.

Security within cloud computing is an especially worrisome issue because of the fact that the devices used to provide services do not belong to the users themselves. The users have no control of, nor any knowledge of, what could happen to their data. This is a great concern in cases when users have valuable and personal information stored in a cloud computing service. Personal information could be "leaked" out, leaving an individual user or business vulnerable for attack. Users will not compromise their privacy so cloud computing service providers must ensure that the customers' information is safe. This, however, is becoming increasingly challenging because as security developments are made, there always seems to be someone to figure out a way to disable the security and take advantage of user information. Aware of these security concerns, cloud computing service providers could possibly face extinction if problems which are hindering fail-proof security are not resolved.

Designing and creating error-proof and fail-proof security for cloud computing services falls on the shoulders of engineers. These engineers face many challenges. The services must eliminate risk of information theft while meeting government specifications. Some governments have laws limiting where personal information can be stored and sent. If too much security detail is relayed to the users, then they may become concerned and decide against cloud computing altogether. If too little security detail is relayed to the users, then the customers will certainly find business elsewhere. If a cloud computing service provider's security is breached and valuable user information is gathered and used against them, then the user could, and likely would, sue the provider and the company could lose everything. Given these scenarios, engineers certainly have to be flawless in their security mechanism designs.

Engineers cannot fixate only on the present issues with cloud computing security but they must also prepare for the future. As cloud computing services become increasingly popular, new technologies are destined to arise. For example, in the future, user information is likely going to be transferred among different service providers. This certainly increases the complexity of the security mechanism because the information must be tracked and protected wherever it may go. As businesses begin to rely upon cloud computing, the speed of services will need to be as high as possible. However, the speed of services cannot come about at the cost of lowered security, after all security is the top concern among cloud computing users. Another aspect of future cloud computing that engineers must be aware of is the increasing population of cloud computing users. As cloud computing services become more popular, the cloud will have to accommodate more users. This raises even more concern about information security and also increases the complexity of cloud computing security mechanisms. Security designers must track every bit of data and also protect information from other users as well as provide a fast and efficient service.

Risk of information theft is different for every cloud computing service and thus should be addressed differently when designing security schemes. For example, services which process and store public information that could also be found in say, a newspaper, would need a very low amount of security. On the other hand, services which process personalized data about an individual or a business must be kept confidential and secure. This type of data is most often at risk when there is unauthorized access to the service account, lost copies of data throughout the cloud, or security faults. Because of these concerns, a security scheme in which all data have a "tracking device" that is limited to certain locations could be implemented. However, care must be taken to also ensure that the data is not transferred or processed in any way without the consent and knowledge of the user.

When preparing to design a cloud computing service and in particular a security scheme, several things must be kept in mind in order to effectively protect user information. First and foremost, protection of user personal information is the top priority. Keeping this in mind

will certainly provide a better service to customers. The amount of valuable personal information that is given to a provider should be kept to a minimum. For example, if the service being provided does not require a telephone number, then the user should not have to offer that information to the provider. Where possible, personal information could be encrypted or somehow kept anonymous. Also, mechanisms which destroy personal data after use and mechanisms which prevent data copying could be implemented. The amount of control that a user has over their information should be increased to a maximum. For example, if a service requires that some personal information be sent out through the cloud and processed, then the user should be able to control where and how that information is sent. The user should also be able to view their data and decide whether or not certain information can be sent out through the cloud. Before any data processing or storing is done, the user should be asked for consent to carry out the operation. Maximizing customer control and influence will earn user trust and establish a wider consumer base.

Some organizations have been focusing on security issues in the cloud computing. The Cloud Security Alliance is a non-profit organization formed to promote the use of best practices for providing security assurance within Cloud Computing, and provide education on the uses of Cloud Computing to help secure all other forms of computing [37]. The Open Security Architecture (OSA) is another organizations focusing on security issues. They propose the OSA pattern [38], which pattern is an attempt to illustrate core cloud functions, the key roles for oversight and risk mitigation, collaboration across various internal organizations, and the controls that require additional emphasis. For example, the Certification, Accreditation, and Security Assessments series increase in importance to ensure oversight and assurance given that the operations are being "outsourced" to another provider. System and Services Acquisition is crucial to ensure that acquisition of services is managed correctly. Contingency planning helps to ensure a clear understanding of how to respond in the event of interruptions to service delivery. The Risk Assessment controls are important to understand the risks associated with services in a business context.

Regarding practical cloud computing products, Microsoft Azure cloud computing plans to offer a new security structure for its multi-tenant cloud environments as well as private cloud software. Google has a multi-layered security process protocol to secure cloud data. Such a process has been independently verified in a successful third-party SAS 70 Type II audit. Google is also able to efficiently manage security updates across our nearly homogeneous global cloud computing infrastructure. Intel uses SOA Expressway Cloud Gateway to enable cloud security. Intel's Cloud Gateway acts as a secure broker that provides security cloud connectors to other companies' cloud products.

3.2. Encryption-on-Demand [26]

The basic idea for encryption-on-demand is that they try to use the encryption-on-demand server which can provide some kind of encryption service. For example, when the server gets a request from user, the website or server will make a unique encryption program, the so-called 'client' program, send a package including such program to the client. However, how to identify a client program or how to assign a client program to a client? They use a tailoring identification (TID) The user can forward the TID to the server, the server will use TID to send them the so-tailor made copy. So how do the users or communication partners compromise the TID? So far, there are many different ways to do that, by using cell phone, text message, or other communication tools. By using some reference information, such as birth city, birth day, or some other personal information all of which can only be known by the communication partners. As long as the two partners can receive the client package including the client program, they can encrypt their messages or email easily. After communication, both users should dump the client program they used. So, every time when they want to communicate with each other privately, they need to download the package.

In Figure 6, the client (both communicating partners) sends the UID (same UID for two partners) to the server, the server will assign a package which binds the encryption system and choose key to the client. Further, the client can use such encryption-on-demand package to

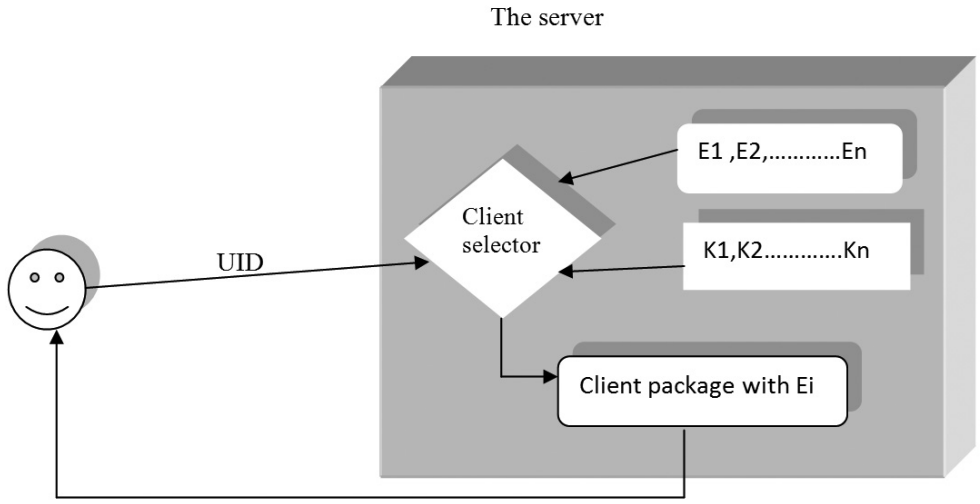

Figure 6. The client sending the UID to the server [26].

encrypt the message and communicate with the partner.

3.3. Security for the Cloud Infrastructure: Trusted Virtual Data Center Implementation [27, 28]

For managing the hardware or resources in the same physical system, the VMM (virtual machine monitor) can create multiple VMs (virtual machines). For the server, this infrastructure provides a very convenient way to create, migrate, and delete the VMs. The cloud computing concept can easily achieve the big scale and cheap services. However, using one physical machine to execute all the workloads of all the users, it will make some serious security issues. Because this infrastructure needs many likelihood components, it will easily lead to the misconfiguration problems. The so-called trusted virtual data center has different VMs and associated hardware resources. In this way, the VM will know which resource it can access. TVDc can separate each customer workloads to different associated virtual machines. The advantages are very obvious, 1) make sure no workload can leak to other customers, 2) in case of some malicious programs like viruses, they cannot be spread to other nodes, 3) prevent the misconfiguration problems.

TVDc uses the so-called isolation policy, so it can separate both the hardware resource and users' workload or data. The isolation policy can manage the data center, access the VMs, and switch from one VM to another VM. TVDc consists of a bunch of VMs and the associated resources which can be used for one user's program. In TVD, the VMs have some labels which can be identified uniquely. For example, one label is corresponded with one customer or the customer's data. TVDc isolation policy includes two major tasks: (1) label that can be used to indentify the VMs which are assigned to the customers, (2) allow all the VMs to run on the same TVD. Based on the security label, the control management can assign the authority to the users for the accessibility. Figure 7 illustrates such a principle.

There are three basic components in the network (Figure 8), such as Client: the laptop, desktop or PDA which can be seen as the data resource. These data need to be processed by the cloud servers. Cloud storage server (CSS) has a huge space to store the data. Third party auditor (TPA) can be responded with the data verification. In Figure 8, the client can deliver the data to the cloud server, the admin will process the data before storage. Since all these data aren't processed in the local computer, the customer needs to verify whether or not the data is correct. Without monitoring the data, the user needs a delegate like the TPA to monitor the data.

The client needs to check the data in the server and make sure all the data of the cloud is correct without any modifying periodically. However, in reality, we assume the adversary can freely access the storage in the server. For example,

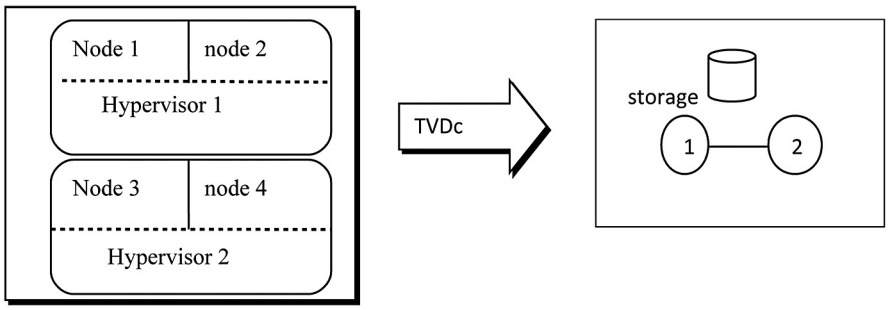

Figure 7. Enabling public verifiability and data dynamics for storage security [27].

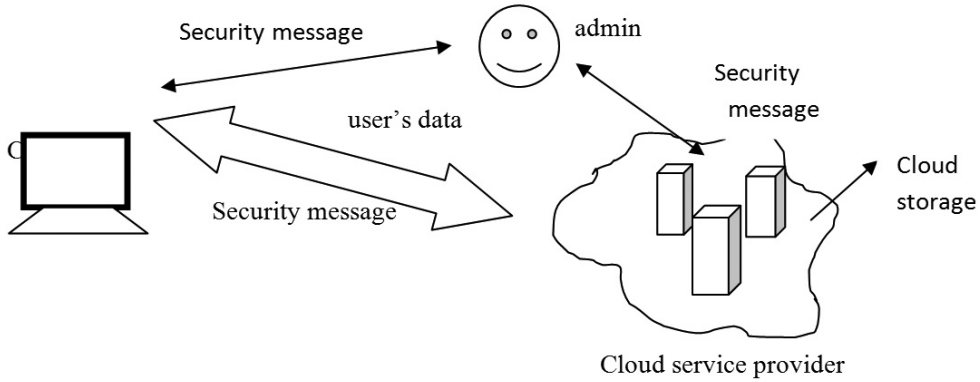

Figure 8. The architecture of cloud data storage [27].

the adversary can play a role of a monitor or TPA. In this way, the adversary can easily cheat the user during the verifying of the data's correctness. Generally, the adversary can use some methods to generate the valid acknowledgments and deliver the verification. Both the server and user can hardly detect such attack.

3.4. Towards Trusted Cloud Computing [28]

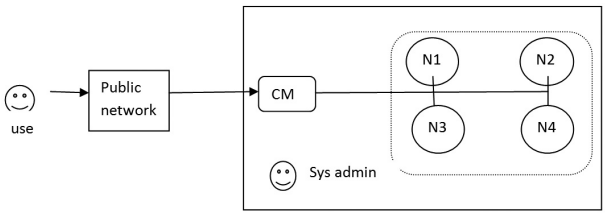

Figure 9. The simple architecture of Eucalyptus [28].

A traditional trusted computing architecture can provide some-degree security for the customers. Such system can forbid the owner of a host to interfere all the computation. The customer can also run the remote testing program which can let the customer know if the procedure of the host is secure or not. If the users or customers detect any kind of abnormal behavior from the host, they can immediately terminate their VM-machines. Unfortunately, such apparent perfect platforms also have some fatal flaws. For example, as we know, the service providers always provide a list of available machines to the customers. Afterwards, the customer will be automatically assigned a machine. However such dynamical assigned machine from the provider backend can incur some kinds of security threat which cannot be solved by such system.

Many cloud providers allow customers to access virtual machines which were hosted by service providers. So user or customer can be seen as the data source for the software running in the VM in the lower layer. In contrast, in the higher layer, the server side can provide all the application on-demand.

The reason of the difficulty to provide an effective security environment for the user is the fact that all the data which need to be processed will be executed directly at higher layers. Briefly,

we just focus on the lower layer in the customer side, because it is managed more easily.

In Eucalyptus (Figure 9), this system can manage multiple clusters in which every node can run a virtual machine monitor which can be used to host the user's VM. CM is the cloud manager which responds to a set of nodes in one cluster. From the user side, Eucalyptus can provide a bunch of interfaces to use the VM which is assigned to the customer. Every VM needs a virtual machine image for launching it. Before launching a VM, VM needs to load VMI. For CM, it can provide some kind of services which can be used to add or remove VMI or users.

The sys admin in the cloud has some kind of privilege and ability to use the backend perform various kind of attacks such as access to the memory of one user's VM. As long as the sys admin has the root privileges at the machine, he or she can use some special software to do some malicious action. Xen can do the live migration, switching its physical host. Because Xen is running at the backend of the provider, Xenaccess can enable the sys admin to run as a customer level process in order to directly access the data of a VM's memory. In this way, the sys admin can do more serious attacks such as the cold boot attacks. Currently we don't worry about such vulnerability because most of IaaS (infrastructure as a Service) providers have some very strict limitation in order to prevent one single person to accumulate all the authorities. Such policy can efficiently avoid the physical access attacks.

We assume the sys admins can log in any machine by using the root authority. In order to access the customers' machine, the sys admins can switch a VM which already runs a customer's VM to one under his or her control. Thus, the TCCP limit the execution of VM in the IaaS perimeter and sys admin cannot access the memory of a host's machine running a VM.

The Trusted Computing (Figure 10) is based on a smart design which uses a so-called trusted platform module (TPM) chip which has a kind of endorsement private key (EK). Moreover, each TPM is bundled with each node in the Perimeter. The service providers can use public key to ensure both the chip and the private key are correct.

At the boot time, the host' machine will generate a list of ML which include a bunch of hashes which is so-called the BIOS, further, both the boot loader and the software begin to run the platform. ML will be stored in the TPM of a host. In order to test the platform, the platform will ask the TPM in the host machine to make a message including the ML, a random number K, and the private EK. Moreover, the sender will send this message to the remote part. In the remote part, by using the public key, it can easily decrypt the message and authenticate the

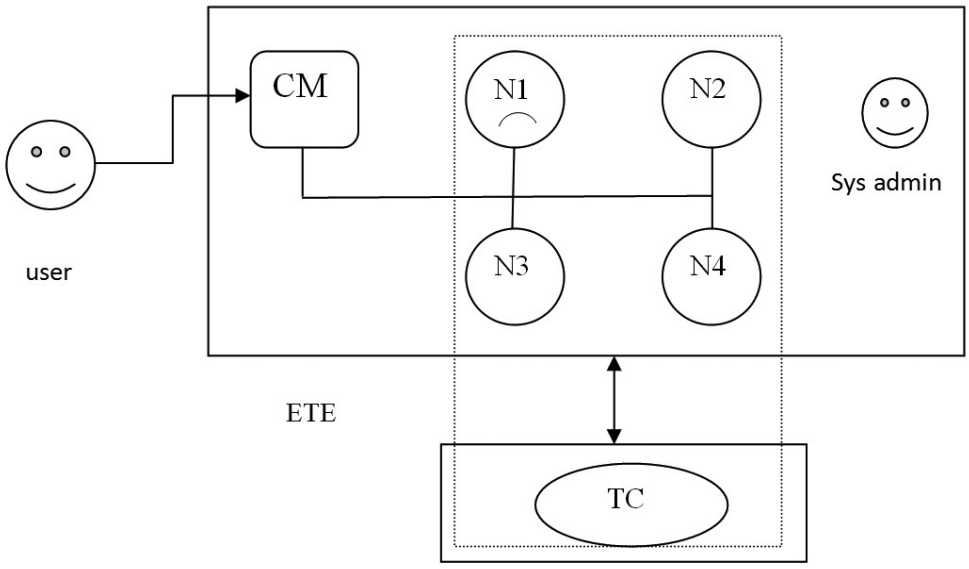

Figure 10. The trusted cloud computing platform TC (trusted coordinator) [28].

correspond host. Here, we use a table to represent the major ideas and differences for the above 3 security methods.

Table 3 compares the above three security schemes.

3.5. Privacy Model [29]

Privacy is a fundamental human right; security schemes in cloud computing services must meet many government regulations and abide by many laws [29]. These laws and regulations are primarily aimed toward protecting information which can be used to identify a person (such as a bank account number or social security number). Of course, these stipulations make cloud computing security even more difficult to implement. Many security schemes have been proposed, but the factor of accountability must be included in all systems, regardless of specific components.

Reasonable solutions to cloud computing security issues involve several elements. First, users must constantly be informed of how and where their data is being sent. Likewise, it is just as important to inform users of how and where their incoming information is received. Second, service providers should give users contractual agreements that assure privacy protection. This type of accountability will provide a control mechanism which will allow users to gain trust of their providers. Third, service providers must accept a responsibility to their customers to establish security and privacy standards. Having standards will certainly help to provide some organization to all security schemes. Lastly, service providers should work with their customers to achieve some feedback in order to continuously update and improve their security schemes.

It is very obvious that cloud computing will greatly benefit consumers as well as internet providers. In order for cloud computing applications to be successful however, customer security must be attained. Presently, the security of cloud computing users is far from certain. Cloud computing security must be greatly improved in order to earn the trust of more people who are potential cloud computing customers. One tempting solution, which has been focused on in the past, is to hide user identifiable information and provide means of anonymous data transfer between user and a cloud computing service. However, this solution is not the best way to provide cloud computing security because users often need to communicate identifiable information to the cloud computing service provider in order to receive the requested service. A concept will now be presented in the context of a group or business participating in a cloud computing service [29].

One suggestion for information privacy in cloud computing services is not to hide identifiable data tracks, but rather to encrypt user identifiable data tracks. This particular method is based on a vector space model which is used to represent group profiles as well as group member profiles. Suppose that a group, consisting

Security method of cluster computing	The major ideas for the methods	pros	cons
Encryption-on-Demand	Both sender and receiver share the same TID in order to get client package for encryption	Using random encrypt-system good for security	Need local machine to encrypt the data
Security for the cloud infrastructure: Trusted virtual data center implementation	By using TVDc, the server side can easily assign the different components for different users preventing the data leak to other users	Reduce the payload for one physical machine and avoid the misconfiguration problem, good for users data security	The user should check the data frequently to make sure it won't be changed
Towards Trusted Cloud Computing	The server can generate a random number K and a private key sent to the user for authentication	Good security by using private key and public key	It can also have some problems if the attackers use the playback attack

Table 3. The major ideas and differences for 3 security methods [29].

of several members, is participating in information exchange via a cloud computing service. When the members are logged on to the service, any information that each member transfers is continuously tracked. While all information is being tracked, false information is continuously generated. The actual information and generated false information are randomly mixed together. This information mixture disrupts the ability of any potential hacker to obtain user identifiable information. The false information generated is not completely random, but rather it is constructed to be similar to the actual information in order to further deceive any potential hacker. Furthermore, the generated false data is cleverly constructed such that is appears to have been generated by the users. This method also allows users to adjust the level of desired privacy. Of course, however, different levels of security have desirable and undesirable tradeoffs (less privacy allows for faster data transfer, etc.).

The vector space model used for this privacy protection method is used to represent user profiles. All information within the vector space model is represented as a vector of weighted terms, where the terms are weighted by level of importance. Suppose some data, d, is represented as an n-dimensional vector $d = (w_1, w_2, w_3, \ldots w_i, \ldots, w_n)$ where w_i represents the weight of the i^{th} term. There are many ways to calculate the weight of a term but often the term's frequency and importance are the main factors. The vector space is updated with every website visit or any other information transfer based upon user preference. For the case above, with group participation, the vector space associated with the group profile is a weighted average of all member vectors. Any new data that could possibly be transferred to a user is analyzed in vector form and if the vector representing this data closely matches the vector representing the user profile, then permission for data transfer is granted. The similarity between some arbitrary data and a user profile, or between two profiles, can be realized by the following relation:

$$S(u_j, u_k) = \frac{\sum_{i=1}^{n} (tu_{ij} \cdot tu_{ik})}{\sqrt{\sum_{i=1}^{n} tu_{ij}^2 \cdot \sum_{i=1}^{n} tu_{ik}^2}} \qquad (4)$$

where u_j and u_k represent the vectors themselves, tu_{ij} and tu_{ik} represent the i^{th} terms in each vector, and n represents the number of terms contained within each vector.

The generator used to create false information is a very important part of the non-anonymous privacy protection method. This generator, called the Transaction Generator, references a database of terms which are closely related to terms associated with group interests and the service being provided. Whenever information is needed to be encrypted, the Transaction Generator uses terms in the database to enter into a search engine. The generator then randomly activates one of the thousands of web sites which result from the search, to generate a complete transfer of false data (false only relative to the user requested or sent information). The number of false transactions generated can be controlled by the user (or group). In the method being described, the number of user-controlled falsified transactions is represented by *Tr*. The Transaction Generator uses this number to generate false transactions each time the user sends out information. The generator does not, however, generate *Tr* falsified transactions for each user information exchange, rather it cleverly generates *Tr* average falsified transactions for each user information exchange. Thus any predictability is eliminated and security is maximized. To guarantee that any potential hackers are distracted, the majority of falsified transactions should consist of web pages which are not directly related to the actual information, but rather to the general idea of the actual information. Otherwise, hackers could accidentally be given a route to find identifiable information. Each time the generator produces a falsified transaction, a vector of weighted terms is built up which can be used to strengthen the user or group profile. An illustration of the transaction generation process is shown in Figure 11.

There are three calculated parameters which are used to update and improve privacy protection each time a user exchanges information and a falsified transaction is made: The Internal Group Profile, the Faked Group Profile, and the External Group Profile. A mechanism called the Group Profile Meter is implemented in this method of non-anonymous privacy protection with a group of users. The Group Profile Meter receives a vector of weighted terms, $V_{t^U}^U$,

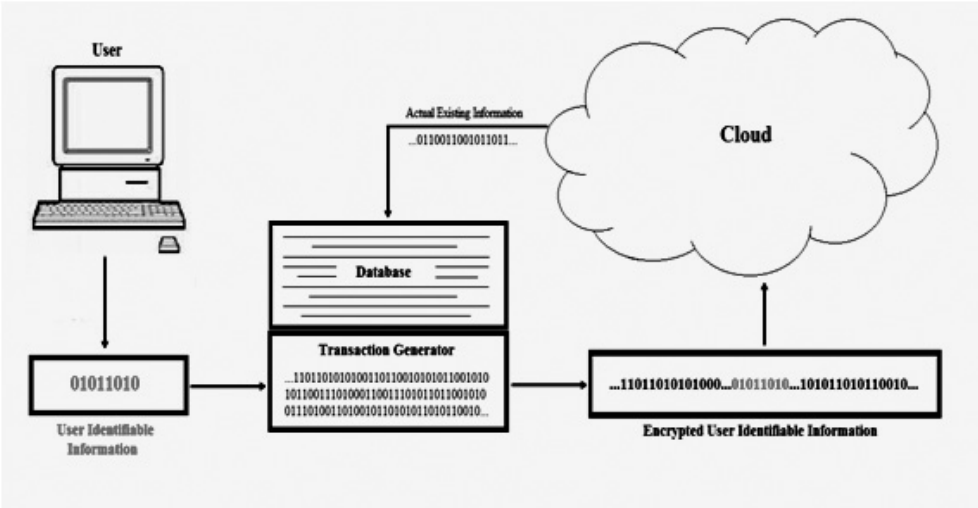

Figure 11. Falsifield transaction generation process for privacy protection [29].

from user information exchange. Using these vectors, a parameter called the Internal Group Profile (IGP) is computed. The Internal Group Profile at any time, t^U, is computed using the following relation:

$$IGP(t^U) = \sum_{i=0}^{\Pr-1} V^U_{t^U-i} \qquad (5)$$

where Pr represents the number of previous vectors in the profile. A similar mechanism called the Faked Group Profile (FGP) is also employed in this privacy protection method. Like the Internal Group Profile, the Faked Group Profile is calculated using information provided by the Group Profile Meter. The Faked Group Profile at any time, t^T, is calculated using the following relation:

$$FGP(t^T) = \sum_{i=0}^{\Pr\times Tr-1} V^T_{t^T-i} \qquad (6)$$

Where Pr and Tr have previously been defined and $V^T_{t^T}$ is a vector of weighted terms received from the Group Profile Meter each time a falsified transaction is made by the Transaction Generator. The Group Profile Meter assembles a parameter called the External Group Profile each time a user or false data vector is received. The External Group Profile (EGP) at any time, t, is simply calculated as follows:

$$EGP(t) = IGP(t^U) + FGP(t^T) \qquad (7)$$

where $EGP(t)$ is updated any time the Internal or External Group Profiles change. Additionally, the Group Profile Meter can compare the Internal and External Group Profiles by calculating a similarity parameter. The similarity between IGP and FGP is calculated by using the following relation:

$$S(IGP, EGP) = \frac{\sum_{i=1}^{n} (tigp_i \cdot tegp_i)}{\sqrt{\sum_{i=1}^{n} tigp_i^2 \cdot \sum_{i=1}^{n} tegp_i^2}} \qquad (8)$$

where $tigp_i$ is the i^{th} term contained within the IGP vector, $tegp_i$ is the i^{th} term contained within the EGP vector, and n is the number of terms contained within each vector. All of the parameters described allow the non-anonymous privacy protection method to be continuously updated and improved in order to provide maximum security for user identifiable information.

3.6. Intrusion Detection Strategy [30]

By now it has become very apparent that cloud computing services can be very beneficial to users. Cloud computing services will allow for better economic efficiency for users as well as service providers. However, maximum efficiency can only be achieved when the occurrence of problems within cloud computing services is at a minimum. Most of these problems

are in the form of security issues. Cloud computing services will not increase in popularity until security issues are resolved. Presently, security within cloud computing is far from perfect and certainly not fail-proof. Cloud computing services will only be practical when users feel safe. One possible solution to safely transfer information is the development of an intrusion detection system [30].

Currently, intrusion detection systems are quite complicated and have much room for improvement. The complexity of these systems arises due to the variety of intrusions which have caused the systems to be built up over time. Cloud computing service providers heavily rely on these intrusion detection systems but at the same time they are faced with the issue of trading good security with efficient performance of the service. Due to the ongoing efforts of hackers, the intrusion detection systems must be often updated and re-worked. One proposal for improvement upon intrusion detection systems will now be presented. This system will be beneficial in detecting intrusions within cloud computing services and will easily adapt to ever-changing attacks.

One proposal for an intrusion detection system is based on a strategy implementing a statistical model for security. This system is designed to be flexible and adaptable to the efforts of attackers. Characteristics of the network involved in the cloud computing service are expressed as some random variables. The value range of these variables is limited in order to be able to identify attacks. When a character, and thus a random variable, changes, then an attack is being attempted. When the values of variables are found to be outside the range of the intervals, then an intrusion has occurred. If the change in time can also be realized, then the intrusion can be successfully detected. This system is organized by simulations implementing trace data. All attacks are arranged in a sample space, Ω. The sample space is then divided into smaller sub-sets. These sub-sets make the system more easily manageable. The set is broken down into mutually exclusive sets. This method will allow the intrusion detection algorithm to be much more efficient. An index for a series of characteristics can be represented by X_i, where X_i is a vector containing p vectors and p is always greater than 1. Intrusion detection can then be realized by the following change-point relation:

$$X_i \approx \begin{cases} N_p(\mu_0, \Sigma_0) & i \leq \tau \\ N_p(\mu_1, \Sigma_1) & i > \tau \end{cases} \tag{9}$$

where τ is the position of attacks, μ is the mean value, and Σ is the nonsingular covariance. Each time the system changes (i.e. an attack is made) the deviation can be calculated by the following relation:

$$Y_k = \left[\frac{k(n-k)}{n}\right]^{\frac{1}{2}} (\overline{X}_{0,k} - \overline{X}_{k,n}) \tag{10}$$

where

$$\overline{X}_{0,k} = \frac{1}{k}\sum_{i=1}^{k} X_i \tag{11}$$

and k is the position of the change. The test statistic is represented by T^2, which can be realized by the following relation:

$$T_k^2 = Y_k' W_k^{-1} Y_k, (k = 1, 2, \ldots, n-1) \tag{12}$$

where W_k is defined in the following relation.

$$W_k = \left[\frac{\sum_{i=1}^{k}(X_i - \overline{X}_{0,k})(X_i - \overline{X}_{0,k})'}{(n-2)} + \frac{\sum_{i=k+1}^{n}(X_i - \overline{X}_{k,n})(X_i - \overline{X}_{k,n})'}{(n-2)}\right] \tag{13}$$

The sum of the deviation can be realized by the following relation:

$$T_f^2 = T_{k,\max}^2, (k = 1, 2, \ldots, n-1) \tag{14}$$

For the covariance probability change of mean value, the variable H_0: $\Sigma_o = \Sigma_1$, is used for hypothesis testing. The test statistic then becomes realized by the following relation:

$$G^* = -2\log(\Lambda) = \sum (n_i - 1)\log\frac{|\hat{\Sigma}_i|}{|\hat{\Sigma}_{pooled}|} \tag{15}$$

Where Λ is calculated by the following relation:

$$\Lambda = \frac{|\hat{\Sigma}_0|^{\frac{k}{2}} \times |\hat{\Sigma}_1|^{\frac{n-k}{2}}}{|\hat{\Sigma}|^{\frac{n}{2}}} \tag{16}$$

Where $\hat{\Sigma}$ is the prediction of Σ. If the expected value, as well as the variance, have both been found to change, then $\mu_0 = \mu_1$ and $\Sigma_0 = \Sigma_1$. Although this has been found to perform well in detecting intrusions under experimental analysis, its reliable performance on the market is yet to be confirmed.

3.7. Dirichlet Reputation Model [31]

As cloud computing becomes more popular, more uses for the service are going to be developed. A newly found use for cloud computing is contained within services which require users to transfer information among one another. The fact that the users of most of these services do not know the user to which they are sending information gives rise to many more security concerns than before. Concerns give rise to fear when users must transfer valuable information such as social security numbers, bank account numbers, etc. In order to make cloud computing services appeal more to users, trust must be established between users and service providers [31].

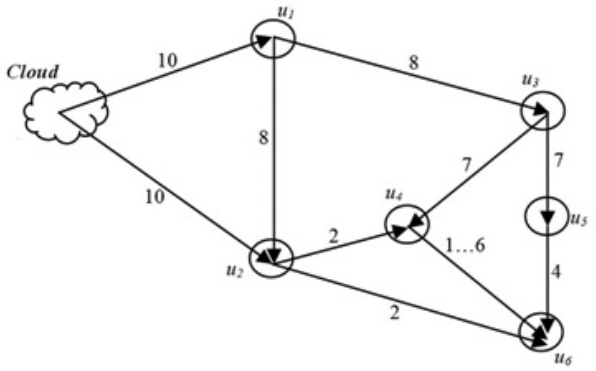

Figure 12. Dirichlet reputation model [32].

One possible solution to resolve the very important security issues is the integration of a system known as the Dirichlet reputation, as shown in Figure 12. This system is implemented in order to observe the behavior of all users participating in a particular service. Basically, users establish trust by interacting with other users in an acceptable manner. If the Dirichlet reputation system discovers that a user is behaving suspiciously or unacceptably, then the user will be punished and possibly not allowed to participate in the cloud computing service. A physical value is assigned to each user which represents their reputation. This value is calculated based upon several factors which are weighted differently and of course can change with user behavior. The reputation value consists of a first-hand reputation and a second-hand reputation. The first-hand reputation is recognized by a direct observation of the user whereas the second-hand reputation is acquired by sharing of the first-hand reputation among other users. Each user participating in the service must report a trust value for each other user with whom they communicate when a second-hand reputation is acquired.

To illustrate the Dirichlet reputation, consider two users, 1 and 2, participating in a cloud computing service. Suppose that user 1 examines some mischievous behavior such as the spread of a virus by user 2. The behavior of user 2 is assumed to follow the Dirichlet distribution with a probability of F_{12}. $Dir(\alpha)$ is designated as the Dirichlet distribution with α representing a vector of real numbers greater than 0. This probability distribution function is used to calculate the first-hand reputation. Bayes' theorem is used to compute this reputation and the typical representation is shown below.

$$\begin{aligned} P(\alpha_i|D) &= \frac{P(D|\alpha_i)P(\alpha_i)}{P(D)} \\ &= Dir(\alpha_i|\alpha_{i1} + N_{i1} \cdots \alpha_{ir_i} + N_{ir_i}) \end{aligned} \tag{17}$$

In the above relation, D represents new data which is provided by user 1 and N represents instantaneous occurrences of new data. The Dirichlet reputation system recognizes only three possible types of behavior: friendly, selfish, and malicious denoted as α_1, α_2, and α_3 respectively. Observed sent or received packets can be realized mathematically by the following two relations

$$\begin{aligned} X &= (X_1, \ldots, X_k) \sim Dir(\alpha) \\ \alpha_0 &= \sum_{i=1}^{k} \alpha_i \end{aligned} \tag{18}$$

where k represents the number of parameters under examination within the distribution function $Dir(\alpha)$ which, in this particular example, is always equal to 3.

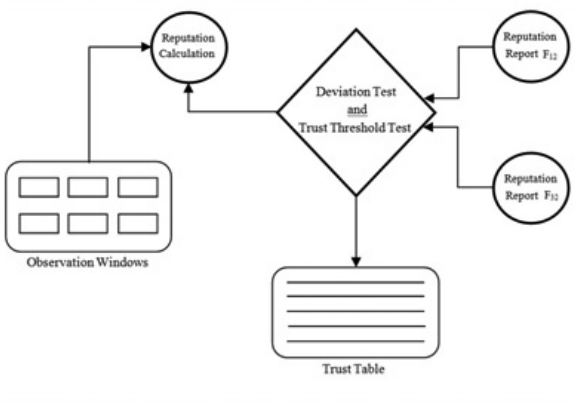

Figure 13. General procedure for calculation of reputation and trust values [32].

In addition to the reputation values calculated using the relations in (14) – (16), the Dirichlet reputation system also implements a trust value (Figure 13). The trust value is used to reflect how trustworthy user reports are of other users. Like the reputation values, the trust values are computed using Bayes' theorem but only two possible outcomes can result, trustworthy or not trustworthy. The probability distribution function used to calculate trust values is a special case of the Dirichlet distribution known as the Beta distribution. The trust value representing the trust that user 1 has for user 2 is denoted as $T_{12} \sim Beta(\gamma, \delta)$ where γ represents trustworthy and δ represents not trustworthy. When users start out in a cloud computing service, $\gamma = 1$ and $\delta = 1$ to establish a starting point. Each time a trustworthy report is made for a user, the trust value for that user changes as $\gamma = \gamma + 1$ and each time a non-trustworthy report is made for a user, the trust value for that user changes as $\delta = \delta + 1$. The trust value that user 1 establishes for user 2 is realized mathematically by the following relation:

$$\omega_{12} = Expectation(Beta(\gamma, \delta)) = \frac{\gamma}{\gamma + \delta} \tag{19}$$

The overall, total reputation value used for evaluation is calculated by blending reports from all users in an environment. An illustration showing the general procedure by which the Dirichlet reputation system implements reputation and trust values is shown in Figure 13.

3.8. Anonymous Bonus Point System [32]

In the current fast-paced world and with cloud computing technologies on the rise, integration of these services with mobile devices has become a necessity [32]. Users now demand mobile services with which they can conduct a business, make a purchase, engage in the stock market, or participate in other services that require valuable information to be transferred. These services operate using complex communication schemes and involve multiple nodes for data transfer. Often, while participating in cloud computing services, users must transfer information to and from individuals or groups whom they do not know. This, of course, presents a situation that is difficult to account for when designing cloud computing security. These types of service systems are especially susceptible to attacks simply because it is easy for attackers to gain access and very difficult for them to be detected and identified. Nonetheless, if designers of these mobile cloud computing services are clever with their security schemes, then attackers can be deterred. A cloud computing service implementing mobile users, along with a proposed security scheme, will now be examined under the context of a personal digital assistant.

Consider a network of three mobile devices (personal digital assistants) capable of communicating with each other over a distance on the order of three hundred to four hundred feet. These three devices will be denoted by *UserA*, *UserB*, and *UserC*. Assuming that all of these devices have access to the cloud, data can be transferred from one user to another on a multihop basis (passing along information one user at a time). This method conserves much energy and prevents the cloud from becoming cluttered. Since users must donate energy from their devices in order to serve other users, each occurrence of a donation results in credit toward the donating user which may later be exchanged for something of monetary value. Now that the basic idea is clear, consider a more complex and more practical situation involving six users of a particular cloud computing service capable of communicating with each other over a distance of several miles. These users will be denoted as u_1, u_2, u_3, u_4, u_5, and u_6. Consider an example situation where user six, u_6, has requested information from the cloud. Figure 13 shows how this information is transferred to u_6 using

a multi-hop communication system. The figure shows all possible paths in this particular network from the cloud to u_6. The numbers between each node represent the credit values passed along to each intermediate user. However, in this particular system credits can only be awarded if the information reaches the intended recipient.

This system certainly has flaws and drawbacks including speed and efficiency, but the most concerning drawback is security. Ordinary cloud services provide means to examine where and who information came from. However, due to high susceptibility to attacks, this particular system keeps this type of information confidential. To achieve confidentiality two primary methods are used. For one method, the sender of information is kept anonymous to every user except the recipient. Still, even then, the recipient must request identification from the sender. For the other method, the sender uses an alias so that all parties involved can know where the information came from, but cannot view the true identity of the sender. Both of these methods prevent others from knowing who is sending information, and thus where to look to find valuable information. Another important security issue involves users' saved data. Often in cloud computing services, users are required to construct profiles where personal information is stored which obviously attracts attackers. This system prevents these kinds of attacks on devices (personal digital assistants in this case) by allocating IP and MAC addresses to each device. This method would at least prevent any unauthorized access to information. In order to compete in the cloud computing market, service providers must put every effort into ensuring a safe and secure network.

3.9. Network Slicing [33]

Although a recent endeavor, cloud computing has proved to be a very useful and very convenient means of providing services. However, as with most other new products, there are many aspects of cloud computing which need to be improved. One particular cloud computing aspect that is of concern is the fact that one single consumer, depending on their service, may have the ability to occupy more than their share of the useful bandwidth. This is obviously a major problem since the cloud computing service providers are trying to make a profit. In developing a scheme to essentially multiplex customers, security becomes more complex and more important. Basically, the task at hand is to conserve our resources, which in this case is the bandwidth of a cloud computing network. A possible solution to this problem proposed here is to utilize virtual machines in order to "slice" the network for bandwidth conservation. This is somewhat of a newly found method and little study and testing have been conducted in this effort. Each user would install a group of virtual machines on their computer, while being able to use a wide range of operating systems. A virtual machine is used for each specific cloud computing service so the number of virtual machines a user must install is dependent upon the number of services they request. The setup of this virtual machine usage is somewhat simple and may be thought of as an information relay. An illustration of this setup is shown in Figure 14.

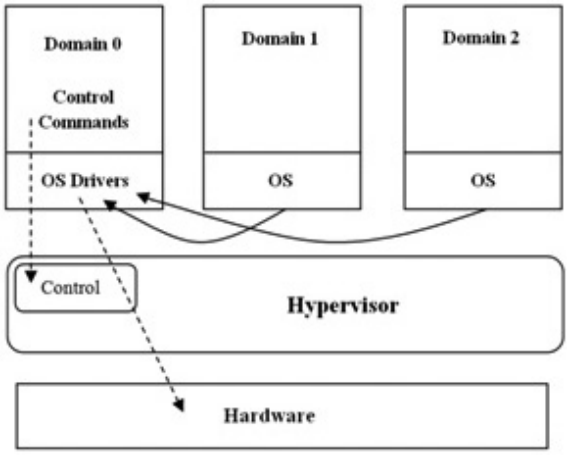

Figure 14. Setup of virtual machine network slicing [33].

The hypervisor serves as a control mechanism which interfaces the operating system and the hardware. The domains represent different users, thus information is passed from one virtual machine to the next. The host driver then communicates with the hardware to perform the requested tasks. It is also suggested to implement a transmission control protocol mechanism. Within the hypervisor, bridges can be used to route information to outside sources. Rather than slicing the network by prioritizing packets, this system slices the network by implementing a schedule. The scheduling scheme

seems to provide a more balanced network. As traffic in the network fluctuates, the scheduler makes adjustments to accommodate these changes. Suppose that a particular set of "n" host users owns a number of virtual machines. Let the following relation represent a vector containing these users.

$$G_i = \{A_1, A_2, A_3, A_4, \ldots, A_n\} \qquad (20)$$

Furthermore, suppose that each user has "m" virtual machines installed on their computer. Let the following equation represent a vector containing these machines.

$$A_i = \{V_1, V_2, V_3, V_4, \ldots, V_m\} \qquad (21)$$

If information is transferred to or from one of the users, then the information would likely travel through $(m-1) * n$ virtual machines. For this reason, security in this system must be near flawless. In order for this network slicing system to be effective, every possible measure must be taken to prevent malicious attacks. There is much work to be done to improve this system, but the payoff would certainly be worth the effort.

3.10. Discussions on Cloud Computing Privacy [29-36]

In the above discussions, five security proposals have been examined and discussed in detail including the effects that each would have on the cloud computing industry. Perhaps the preceding concepts could be used to derive a new method which could implement the strong aspects of each of these fives schemes. Security within cloud computing is far from perfect and has recently become a very puzzling issue to resolve. The complexity with security arises because of constant improvement of attackers' knowledge and accessibility. Before cloud computing services become desirable, customers must feel safe with their information transfer. Each proposal previously under examination has beneficial properties in their own respect. For example, the first model described (the privacy model) implements an economically efficient method while the CP intrusion detection system focuses more effort toward attack prevention. When designing a security scheme for cloud computing services, there underlies a dilemma by which security cannot come at the cost of other desirable aspects such as data speed or affordability. To counter this dilemma, some security schemes like the Dirichlet Reputation system allow the user to control the level of security a great deal. Although some have more than others, all five aforementioned security schemes each have very important and valuable concepts. Table 4 displays a list of the main favorable aspects, or *pros*, of each of the examined five security methods.

To combine the benefits these security proposals have provided, we propose a hybrid security solution as follows. First, we group users into different domains, based on their demands and the services they need. A network slicing hypervisor serves as a control mechanism. We use a vector space model to represent domain profiles as well as domain member profiles. After logged on, the members transfer information which is tracked. Meanwhile, false information is continuously generated and mixed with the tracked information randomly. Security parameters can be customized by the members.

Privacy Model	• Provides very strong encryption of information • Users can easily customize their security parameters • Provides an organized method which can be implemented easily
CP Intrusion Detection	• Protects against a wide variety of intrusion schemes • Provides excellent prevention of attacks
Dirichlet Reputation	• Provides sophisticated system of checks and balances • Avoids ability for attackers to adapt • Provides a great deal of user control
Anonymous Bonus Point	• Best suited for small distances, thus users are well hidden from attackers • Credit rewards provide incentive for users to participate
Network Slicing	• Provides attacker confusion • Conserves network bandwidth • Fast data rates are easily attainable

Table 4. Pros of cloud computing security strategies.

Privacy Model	• Errors and bugs are difficult to find and correct • Services can become bogged down by distracting information • System is only preventative, thus it does not protect against aggressive attackers
CP Intrusion Detection	• Must be updated frequently to confuse attackers • May erroneously detect and terminate non-intrusive information
Dirichlet Reputation	• Relies on complicated strategy that is difficult to implement • User trust yields susceptibility to deceptive customers • Performance is solely dependent upon user participation
Anonymous Bonus Point	• Data speed is drastically reduced • Provides little intrusion protection • Can only be implemented in wireless applications
Network Slicing	• Due to the relay structure, protection is unreliable • Can become expensive when implemented in large networks

Table 5. Cons of cloud computing security strategies.

In information transferring, the system also enhances the ability of encountering attacks by an intrusion detection system based on a strategy implementing a statistical security model with characteristics of the network. For each member, a Dirichlet reputation index is associated. With the index of the receiver, every sender can determine whether to transfer the data or not if the receiver behaves suspiciously. We will further study more details when implementing this hybrid security proposal in our future works.

Just as each security scheme has its own favorable parameters, each scheme also has its own unfavorable parameters, or downfalls. Because no cloud computing security method will ever be free of flaws, an extremely important consideration when designing security schemes must be to balance the favorable aspects and the unfavorable aspects. Yet, this is certainly more complicated than it may seem at first glance. There are a number of factors which must be considered when attempting to balance strong points and downfalls. For example, a system such as the CP intrusion detection strategy, which provides very strong protection against attackers, must also allow the service to be efficient both in performance and in cost. Any security system developed will inevitably exhibit flaws and downfalls, but this does not mean that the system is not usable. Just as users desire some specific parameters included with their security scheme, they also are not concerned about some of the downfalls. For example, a large corporation is likely not to be concerned about the cost of an expensive service if excellent security is included whereas an individual is more likely not to spend as much money on a comparable system. In order to compare strong points to downfalls of the previously mentioned cloud computing security schemes, Table 5 displays a list of the major flaws, or *cons*, included in these systems.

The aforementioned security proposals for implementation into cloud computing are some of the best performing methods in experimental analysis. Although not perfect, and in some cases not even thoroughly tested, these security schemes certainly encompass good solutions to the major issues within cloud computing security. As mentioned before, none of these five security methods under examination exhibit solutions to all problems, but future work may include sub-schemes from each of these methods. Of course one central "template" for developing cloud computing security schemes is highly unlikely, so research must include many different proposals such as the five discussed previously. Cloud computing is a fairly recent endeavor and thus, will require time to develop reasonable and efficient security systems. Security methods within cloud computing will inevitably have to be frequently updated, more so than security implementations for other applications, due to ongoing efforts of attackers. The future of the entire cloud computing industry is essentially dependent upon security schemes to provide privacy and protection of users.

4. Conclusions

Cloud computing will be a major power of the large-scale and complex computing in the feature. In this paper, we present a comprehensive survey on the concepts, architectures, and challenges of cloud computing. We provide introduction in details for architectures of cloud computing in every level, followed by a summary of challenges in cloud computing, in the

aspects of security, virtualization, and cost efficiency. Among them, Security issues are the most important challenge in the cloud computing. We survey comprehensively the security issues and the current methods addressing the security challenges. This survey provides useful introduction on cloud computing to the researchers with interest in cloud computing.

References

[1] G. GRUMAN, E. KNORR, What cloud computing really means. *InfoWorld*, (2009, May). [Online]. Available: `http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031`

[2] L. SIEGELE, Let it rise: A survey of corporate IT. *The Economist*, (Oct., 2008).

[3] P. WATSON, P. LORD, F. GIBSON, Panayiotis Periorellis, and Georgios Pitsilis. *Cloud computing for e-science with carmen*, (2008), pp. 1–5.

[4] R. M. SAVOLA, A. JUHOLA, I. UUSITALO, Towards wider cloud service applicability by security, privacy and trust measurements. *International Conference on Application of Information and Communication Technologies (AICT)*, (Oct., 2010), pp. 1–6.

[5] M.-E. BEGIN, An egee comparative study: Grids and clouds – evolution or revolution. *EGEE III project Report*, vol. 30 (2008).

[6] B. ROCHWERGER, D. BREITGAND, E. LEVY, A. GALIS, K. NAGIN, I. M. LLORENTE, R. MONTERO, Y. WOLFSTHAL, E. ELMROTH, J. CACERES, M. BEN-YEHUDA, W. EMMERICH, F. GALAN, The Reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, vol. 53, no. 4 (July, 2009), pp. 1–11.

[7] L. M. VAQUERO, L. RODERO-MERINO, J. CACERES, M. LINDNER, A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39, 1 (December 2008).

[8] OPENQRM. [Online]: `http://www.openqrm.com`

[9] AMAZON ELASTIC COMPUTE CLOUD (AMAZON EC2) [Online]. Available: `http://aws.amazon.com/ec2/`

[10] L. WANG, G. VON LASZEWSKI, A. YOUNGE, X. HE, M. KUNZE, J. TAO, C. FU, Cloud Computing: A Perspective Study. *New Generation Computing*, vol. 28, no. 2 (2010), pp. 137–146.

[11] J. MURTY, *Programing Amazon Web Services: S3, EC2, SQS, FPS, and SimpleDB*, O'Reilly Media, 2008.

[12] E. Y. CHEN, M. ITOH, Virtual smartphone over IP. In *Proceedings of IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks*, (2010), pp. 1–6.

[13] E. CIURANA, *Developing with Google App Engine*, Spring, 2010.

[14] C. D. WEISSMAN, S. BOBROWSKI, The design of the force.com multitenant internet application development platform. In *Proceedings of the 35th SIGMOD international conference on Management of data*, (SIGMOD '09), (2009) pp. 889–896.

[15] R. T. DODDA, A. MOORSEL, C. SMITH, An Architecture for Cross-Cloud System Management, unpublished.

[16] B. SOTOMAYOR, R. MONTERO, I. LLORENTE, I. FOSTER, Resource Leasing and the Art of Suspending Virtual Machines. In *IEEE International Conference on High Performance Computing and Communications (HPCC09)*, (June 2009), pp. 59–68.

[17] H. CHEN, G. JIANG, A. SAXENA, K. YOSHIHIRA, H. ZHANG, Intelligent Workload Factoring for A Hybrid Cloud Computing Model, unpublished.

[18] M. M. HASSAN, E. HUH, B. SONG, A Framework of Sensor – Cloud Integration Opportunities and Challenges. Presented at the *International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, (January 15-16, 2009), Suwon, South Korea.

[19] K. KEAHEY, T. FREEMAN, Contextualization: Providing One-Click Virtual Clusters. *IEEE International Conference on eScience*, (2008), pp. 301–308.

[20] D. BERNSTEIN, E. LUDVIGSON, K. SANKAR, S. DIAMOND, M. MORROW, Blueprint for the Intercloud – Protocols and Formats for Cloud Computing Interoperability. In *Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services (ICIW '09)*. Washington, DC, USA, pp. 328–336.

[21] J. POWELL, Cloud computing – what is it and what does it mean for education?, unpublished.

[22] SERVER VIRTUALIZATION FAQ [Online]. Available: `http://www.itmanagement.com/faq/server-virtualization/`

[23] K. KEAHEY, Cloud Computing for Science. *Lecture Notes in Computer Science*, vol. 5566 (2009), pp. 478.

[24] A. LENK, M. KLEMS, J. NIMIS, T. SANDHOLM, What's inside the Cloud? An architectural map of the Cloud landscape. *ICSE Workshop on Software Engineering Challenges of Cloud Computing*, (2009), pp. 23–31.

[25] F. TUSA, M. PAONE, M. VILLARI, A. PULIAFITO, CLEVER: A cloud-enabled virtual environment. *Computers and Communications (ISCC), 2010 IEEE Symposium on*, vol., no. (22-25 June 2010), pp. 477–482.

[26] GIDEON SAMID SCHOOL OF ENGINEERING CASE WESTERN RESERVE UNIVERSITY, Encryption-On-Demand: Practical and Theoretical Considerations, 2008.

[27] N. SANTOS, K. P. GUMMADI, R. RODRIGUES, Towards Trusted Cloud Computing.

[28] S. BERGER, R. CA'CERES, K. GOLDMAN, D. PENDARAKIS, R. PEREZ, J. R. RAO, E. ROM, R. SAILER, W. SCHILDHAUER, D. SRINIVASAN, S. TAL, E. VALDEZ, Security for the cloud infrastructure: Trusted virtual data center implementation. 2009.

[29] Y. ELOVICI, B. SHAPIRA, A. MASCHIACH, A New Privacy Model for Hiding Group Intersest while Accessing the Web. In *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society (WPES '02)*, ACM, New York, NY, USA, pp. 63–70

[30] Y. GUAN, J. BAO, A CP Intrusion Detection Strategy on Cloud Computing. In *Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA'09)*, (May 22-24, 2009), Nanchang, P. R. China, pp. 84–87.

[31] L. YANG, A. CEMERLIC, Integrating Dirichlet reputation into usage control. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, (2009), pp. 1–4.

[32] T. STRAUB, A. HEINEMANN, An anonymous bonus point system for mobile commerce based on word-of-mouth recommendation. In *Proceedings of the 2004 ACM symposium on Applied computing (SAC '04)*, (2004) New York, NY, USA, pp. 766–773.

[33] A. NAYAK, B. ANWER, P. PATIL, Network Slicing in Virtual Machines for Cloud Computing. [Online]: `http://www.cc.gatech.edu/projects/disl/`

[34] B. THOMPSON, D. YAO, The Union-Split Algorithm and Cluster-Based Anonymization of Social Networks, 2009.

[35] Q. WANG, C. WANG, J. LI, K. REN, W. LOU, Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing, 2009.

[36] C. WANG, Q. WANG, K. REN, Ensuring Data Storage Security in Cloud Computing.

[37] SECURITY GUIDANCE FOR CRITICAL AREAS OF FOCUS IN CLOUD COMPUTING [Online]. Available: `http://www.cloudsecurityalliance.org/guidance/csaguide.pdf`.

[38] T. MATHER, S. KUMARASWAMY, S. LATIF, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*. O'Reilly Media.

[39] B. SOTOMAYOR, R. S. MONTERO, I. M. LLORENTE, I. FOSTER, Virtual Infrastructure Management in Private and Hybrid Clouds. *IEEE Internet Computing* vol. 13, no. 5 (Sept.-Oct. 2009), pp. 14–22,

[40] J. O. KEPHART, D. M. CHESS, The vision of autonomic computing. *Computer*, 36(1) (2003), pp. 41–50.

[41] L. KLEINROCK, A vision for the internet. *ST Journal of Research*, 2(1) (November 2005), pp. 4–5.

[42] M. MILENKOVIC, S. H. ROBINSON, R. C. KNAUERHASE, D. BARKAI, S. GARG, A. TEWARI, T. A. ANDERSON, M. BOWMAN, Toward internet distributed computing. *Computer*, 36(5) (May 2003), pp. 38–46.

[43] L.-J. ZHANG, EIC Editorial: Introduction to the Body of Knowledge Areas of Services Computing. *IEEE Transactions on Services Computing*, 1(2) (April-June 2008), pp. 62–74.

[44] L. YOUSEFF, M. BUTRICO, D. DA SILVA, Toward a unified ontology of cloud computing. In *Grid Computing Environments Workshop*, (Nov. 2008), pp. 1–10.

[45] L. WANG, J. TAO, M. KUNZE, A. C. CASTELLANOS, D. KRAMER, W. KARL, Scientific Cloud Computing: Early Definition and Experience. In *HPCC '08*, pp. 825–830.

[46] D. CHAPPELL, Introducing the Azure Services Platform. [Online]. `http://download.microsoft.com/download/e/4/3/e43bb484-3b52-4fa8-a9f9-ec60a32954bc/Azure_Services`

[47] M. ARMBRUST, A. FOX, R. GRIFFITH, A. D. JOSEPH, R. KATZ, A. KONWINSKI, G. LEE, D. PATTERSON, A. RABKIN, I. STOICA, M. ZAHARIA, A view of cloud computing. *Commun. ACM*, 53(4) (April 2010), pp. 50–58.

Received: August, 2010
Revised: February, 2011
Accepted: March, 2011

Contact addresses:
Fei Hu
Department of Electrical and Computer Engineering
University of Alabama
Tuscaloosa, AL, USA
e-mail: `fei@eng.ua.edu`

Meikang Qiu
Department of Electrical and Computer Engineering
University of Kentucky
Lexington, KY, USA
e-mail: `mqiu@engr.uky.edu`

Jiayin Li
Department of Electrical and Computer Engineering
University of Kentucky
Lexington, KY, USA
e-mail: `jli6@engr.uky.edu`

Travis Grant
Draw Tylor
Seth McCaleb
Lee Butler
Richard Hamner
Electrical and Computer Engineering
University of Alabama
Tuscaloosa, AL, USA
e-mail: `{tggrant, labutler, rahamner1}@crimson.ua.edu`;
`shmccaleb@gmail.com`

FEI HU is currently an associate professor in the Department of Electrical and Computer Engineering at the University of Alabama, Tuscaloosa, AL, USA. His research interests are wireless networks, wireless security and their applications in biomedicine. His research has been supported by NSF, Cisco, Sprint, and other sources. He obtained his first Ph.D. degree at Shanghai Tongji University, China in Signal Processing (in 1999), and second Ph.D. degree at Clarkson University (New York State) in the field of Electrical and Computer Engineering (in 2002).

MEIKANG QIU received the B.E. and M.E. degrees from Shanghai Jiao Tong University, China. He received the M.S. and Ph.D. degrees of Computer Science from the University of Texas in Dallas in 2003 and 2007, respectively. He had worked at Chinese Helicopter R&D Institute and IBM. Currently, he is an assistant professor of ECE at the University of Kentucky. He is an IEEE Senior member and has published 100 papers. He has been on various chairs and TPC members for many international conferences. He served as the Program Chair of IEEE EmbeddCom'09 and EM-Com'09. He received Air Force Summer Faculty Award 2009 and won the best paper award in IEEE Embedded and Ubiquitous Computing (EUC) 2009. His research interests include embedded systems, computer security, and wireless sensor networks.

JIAYIN LI received the B.E. and M.E. degrees from Huazhong University of Science and Technology (HUST), China, in 2002 and 2006, respectively. Now he is pursuing his Ph.D. degree in the Department of Electrical and Computer Engineering (ECE), University of Kentucky. His research interests include software/hardware co-design for embedded system and high performance computing.

TRAVIS GRANT, DRAW TYLOR, SETH MCCALEB, LEE BUTLER, AND RICHARD HAMNER are students in the Department of Electrical and Computer Engineering at the University of Alabama, Tuscaloosa, AL, USA.