TRUST MODEL FOR RELIABLE FILE EXCHANGE IN CLOUD COMPUTING

[1]Edna Dias Canedo, Rafael Timóteo de Sousa Junior, and Robson de Oliveira Albuquerque

[1]Electrical Engineering Department, University of Brasília – UNB – Campus Darcy Ribeiro – Asa Norte – Brasília – DF, Brazil, 70910-900
ednacanedo@unb.br,robson@redes.unb.br,desousa@unb.br

ABSTRACT

The recent developments in cloud computing technology show an increase in security, privacy and trust related problems, in many ways, which haven't been predicted by the ones who have been designing clouding environments. Among them, the users trust problem and cloud computing resources secure access warranty have deserved special attention. At this study, it will be presented a review on the concepts of cloud computing, trust and reputation, and some open questions related to trust and security in cloud computing environments will be discussed. Computing systems trust and reputation representation have been widely discussed and applied in a lot of information technology scenarios, becoming subject of scientific researches both from a theoretical and practical point of view. This paper proposes the development of a high level trust model to ensure a reliable files exchange among users in a private cloud, as well as the calculation process of trust among these users, according to the metrics previously established.

KEYWORDS

Availability, Cloud Computing, Distributed Computing, Filesystem, Integrity, Reputation, Security and Trust

1. INTRODUCTION

The widespread use of Internet connected systems and distributed applications has triggered a revolution towards the adoption of pervasive and ubiquitous cloud computing environments. These environments allow users and clients to purchase computing power according to necessity, elastically adapting to different performance needs while providing higher availability. Several web-based solutions, such as Google Docs and Customer Relationship Management (CRM) [1] applications, now operate in the software as a service model. Much of this flexibility is made possible by virtual computing methods, which can provide adaptive resources and infrastructure in order to support scalable on-demand sales of such applications. Virtual computing is also applied to stand-alone infrastructure as a service solution, such as Amazon Elastic Cloud Computing (EC2) and Elastic Utility Computing Architecture Linking Your Programs to Useful Systems (Eucalyptus) [1].

As a result, the cloud computing frameworks and environments are able to address different issues in current distributed and ubiquitous computing systems. The availability of infrastructure as a service and platform as a service environments provided a fundamental base for building cloud computing based applications. It also motivated the research and development of technologies to support new applications. As several large companies in the communications and information technology sector have adopted cloud computing based applications, this approach is becoming a de facto industry standard, being widely adopted by different organizations.

DOI : 10.5121/ijcsit.2012.4101

Since the adoption of the cloud computing paradigm by IBM Corporation around the end of 2007, other companies such as Google (Google App Engine), Amazon (Amazon Web Services (AWS), EC2 (Elastic Compute Cloud) and S3 (Simple Storage Service)), Apple (iCloud) and Microsoft (Azure Services Platform) have progressively embraced it and introduced their own new products based on cloud computing technology [2]. However, cloud computing still poses risks related to data security in its different aspects (integrity, confidentiality and authenticity).

Cloud computing provides a low-cost, scalable, location independent infrastructure for data management and storage. The rapid adoption of Cloud services is accompanied by increasing volumes of data stored at remote servers, so techniques for saving disk space and network bandwidth are needed. A central up and coming concept in this context is deduplication, where the server stores only a single copy of each file, regardless of how many clients asked to store that file. All clients that store the file merely use links to the single copy of the file stored at the server. Moreover, if the server already has a copy of the file, then clients do not even need to upload it again to the server, thus saving bandwidth as well as storage (this is termed client-side deduplication).

In a typical storage system with deduplication, a client first sends to the server only a hash of the file and the server checks if that hash value already exists in its database. If the hash is not in the database then the server asks for the entire file. Otherwise, since the file already exists at the server (potentially uploaded by someone else), it tells the client that there is no need to send the file itself. Either way the server marks the client as an owner of that file, and from that point on the client can ask to restore the file (regardless of whether he was asked to upload the file or not).

The client-side deduplication introduces new security problems. For example, a server telling a client that it need not send the file reveals that some other client has the exact same file, which could be sensitive information. A malicious client can use this information to check whether specific files were uploaded by other users, or even run a brute force attack which identifies the contents of certain fields in files owned by other users, by trying to upload multiple variants of the same file which have different values for that field. The findings apply to popular file storage services such as MozyHome and Dropbox, among others.

In this paper, we review the main cloud computing architecture patterns and identify the main issues related to security, privacy, trust and availability. In order to address such issues, we present a high level architecture for trust models in cloud computing environments.

This paper is organized as follows. In Section II, we present an overview of cloud computing, presenting a summary of its main features, architectures and deployment models. In Section III, we present review some related work about security, file system and trust in the cloud. In section IV is presented the file distribution in cloud. In section V, we introduce the proposed trust model. Finally, in Section VI, we conclude with a summary of our results and directions for new research.

2. CLOUD COMPUTING

Cloud computing refers to the use, through the Internet, of diverse applications as if they were installed in the user's computer, independently of platform and location. Several formal definitions for cloud computing have been proposed by industry and academia. We adopt the following definition: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [3]. This definition includes cloud architectures, security, and deployment strategies.

Cloud computing is being progressively adopted in different business scenarios in order to obtain flexible and reliable computing environments, with several supporting solutions available in the market. Being based on diverse technologies (e.g. virtualization, utility computing, grid computing and service oriented architectures) and constituting a whole new computational paradigm, cloud computing requires high level management routines. Such management activities include: (a) service provider selection; (b) virtualization technology selection; (c) virtual resources allocation; (d) monitoring and auditing in order to guarantee Service Level Agreements (SLA).

Computational trust can be leveraged in order to establish architecture and a monitoring system encompassing all these needs and still supporting usual activities such as planning, provisioning, scalability and security. Chang et al. [4] present a few challenges related to security, performance and availability in the cloud.

2.1. Cloud Computing Architecture

Cloud computing architecture is based on layers. Each layer deals with a particular aspect of making application resources available. Basically there are two main layers: a lower and a higher resource layer. The lower layer comprises the physical infrastructure and is responsible for the virtualization of storage and computational resources. The higher layer provides specific services. These layers may have their own management and monitoring system, independent of each other, thus improving flexibility, reuse and scalability. Figure 1 presents the cloud computing architectural layers [5] [30].

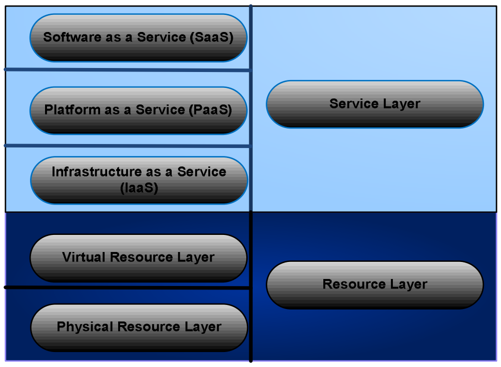
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)
Service Layer
Virtual Resource Layer
Physical Resource Layer
Resource Layer

Figure 1. Cloud Computing Architecture [5] [30]

2.2. Software as a Service

Software as a Service (SaaS) provides all the functions of a traditional application, but provides access to specific applications through Internet. The SaaS model reduces concerns with application servers, operating systems, storage, application development, etc. Hence, developers may focus on innovation, and not on infrastructure, leading to faster software systems development.

SaaS systems reduce costs since no software licenses are required to access the applications. Instead, users access services on demand. Since the software is mostly Web based, SaaS allows better integration among the business units of a given organization or even among different software services. Examples of SaaS include [1]: Google Docs and Customer Relationship Management (CRM) services.

2.3. Platform as a Service

Platform as a Service (PaaS) is the middle component of the service layer in the cloud. It offers users software and services that do not require downloads or installations. PaaS provides an infrastructure with a high level of integration in order to implement and test cloud applications. The user does not manage the infrastructure (including network, servers, operating systems and storage), but he controls deployed applications and, possibly, their configurations [2].

PaaS provides an operating system, programming languages and application programming environments. Therefore, it enables more efficient software systems implementation, as it includes tools for development and collaboration among developers. From a business standpoint, PaaS allows users to take advantage of third party services, increasing the use of a support model in which users subscribe to IT services or receive problem resolution instructions through the Web. In such scenarios, the work and the responsibilities of company IT teams can be better managed. Examples of SaaS [1] include: Azure Services Platform (Azure), Force.com, EngineYard and Google App Engine.

2.4. Infrastructure as a Service

Infrastructure as a Service (IaaS) is the portion of the architecture responsible for providing the infrastructure necessary for PaaS and SaaS. Its main objective is to make resources such as servers, network and storage more readily accessible by including applications and operating systems. Thus, it offers basic infrastructure on-demand services. IaaS has a unique interface for infrastructure management, an Application Programming Interface (API) for interactions with hosts, switches, and routers, and the capability of adding new equipment in a simple and transparent manner. In general the, user does not manage the underlying hardware in the cloud infrastructure, but he controls the operating systems, storage and deployed applications. Eventually he can also select network components such as firewalls.

The term IaaS refers to a computing infrastructure, based on virtualization techniques that can scale dynamically, increasing or reducing resources according to the needs of applications. The main benefit provided by IaaS is the pay-per-use business model [2]. Examples of IaaS [1] include: Amazon Elastic Cloud Computing (EC2) and Elastic Utility Computing Architecture Linking Your Programs To Useful Systems (Eucalyptus).

2.5. Roles in Cloud Computing

Roles define the responsibilities, access and profile of different users that are part of a cloud computing solution. Figure 2 presents these roles defined in the three service layers [5].

The provider is responsible for managing, monitoring and guaranteeing the availability of the entire structure of the cloud computing solution. It frees the developer and the final user from such responsibilities while providing services in the three layers of the architecture.

Developers use the resources provided by IaaS and PaaS to provide software services for final users. This multi-role organization helps to define the actors (people who play the roles) in cloud computing environments. Such actors may play several roles at the same time according to need or interest. Only the provider supports all the service layers.

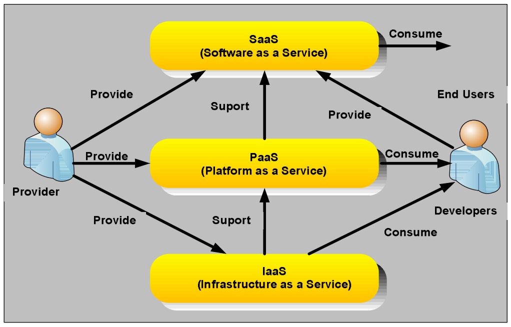

Figure 2. Roles in cloud computing [5]

2.6. Cloud Computing Deployment

According to the intended access methods and availability of cloud computing environments, there are different models of deployment [6] [30]. Access restriction or permission depends on business processes, the type of information and characteristics of the organization. In some organizations, a more restrict environment may be necessary in order to ensure that only properly authorized users can access and use certain resources of the deployed cloud services. A few deployment models for cloud computing are discussed in this section. They include private cloud, public cloud, community cloud and hybrid cloud, which are briefly analyzed in Table 1.

Table 1. Models of deployment of cloud services [6] [30]

Cloud Model	**Description**
Private	In this model, the cloud infrastructure is exclusively used by a specific organization. The cloud may be local or remote, and managed by the company itself or by a third party. There are policies for accessing cloud services. The techniques employed to enforce such private model may be implemented by means of network management, service provider configuration, authorization and authentication technologies or a combination of these.
Public	Infrastructure is made available to the public at large and can be accessed by any user that knows the service location. In this model, no access restrictions can be applied and no authorization and authentication techniques can be used.
Community	Several organizations may share the cloud services. These services are supported by a specific community with similar interests such as mission, security requirements and policies, or considerations about flexibility. A cloud environment operating according to this model may exist locally or remotely and is normally managed by a commission that represents the community or by a third party.
Hybrid	Involves the composition of two or more clouds. These can be private, community or public clouds which are linked by a proprietary or standard technology that provides portability of data and applications among the composing clouds.

3. Related Works

This section review some related work about security, file system and trust in the cloud.

3.1. Security in the Cloud

A number of technologies have been employed in order to provide security for cloud computing environments. The creation and protection of security certificates is usually not enough to ensure the necessary security levels in the cloud. Cryptographic algorithms used with cloud applications usually reduce performance and such reduction must be restricted to acceptable levels [7].

Cloud computing offers users a convenient way of sharing a large quantity of distributed resources belonging to different organizations. On the other hand, the very nature of the cloud computing paradigm makes security aspects quite more complex. Trust is the main concern of consumers and service providers in a cloud computing environment [8]. The inclusion of totally different local systems and users of quite diverse environments brings special challenges to the security of cloud computing. On one hand, security mechanisms must offer users a high enough level of guarantees. On the other hand, such mechanism must not be so complex as to make it difficult for users to use the system. The openness and computational flexibility of popular commercially available operating systems have been important factors to support the general adoption of cloud computing. Nevertheless, these same factors increase system complexity, reduce the degree of trust and introduce holes that become threats to security [8].

Huan et al. [9] investigate the different security vulnerability assessment methods for cloud environments. Experiments show that more vulnerabilities are detected if vulnerable tools and servers are in the same LAN. In other word, the hackers can find an easier way to get the target information if it is on the same LAN of compromised systems. Experimental results can be used to analyze the risk in third party compute clouds. Popovic et al. [10] discuss security issues, requirements and challenges that Cloud Service Providers (CSP) face during cloud engineering. Recommended security standards and management models to address these are suggested both for the technical and business community.

3.2. Filesystem Security

As the number of devices managed by users is continually increasing, there is a growing necessity of synchronizing several hierarchically distributed file systems using ad-hoc connectivity.

Uppoor et al. [11] present a new approach for synchronizing of hierarchically distributed file systems. Their approach resembles the advantages of peer-to-peer synchronization, storing online master replicas of the shared files. The proposed scheme provides data synchronization in a peer-to-peer network, eliminating the costs and bandwidth requirements usually present in cloud computing master-replica approaches.

The work in [12] presents CDRM, a scheme for dynamic distribution of file replicas in a cloud storage cluster. This scheme periodically updates the number and location of file block replicas in the cluster. The number of replicas is updated according to the actual availability of cluster nodes and the expected file availability. The dynamic distribution algorithm for replica placement takes into account the storage and computational capacity of the cluster nodes, as well as the bandwidth of the communication network. An implementation of the proposed scheme using an open source distributed file system named HDFS (Hadoop Distributed File System) is discussed. Experimental measurements point out that the dynamic scheme outperforms existing static file distribution algorithms.

3.3. Trust

The concepts of trust, trust models and trust management have been the object of several recent research projects. Trust is recognized as an important aspect for decision-making in distributed and auto-organized applications [13], [14]. In spite of that, there is no consensus in the literature on the definition of trust and what trust management encompasses. In the computer science literature, Marsh is among the first to study computational trust. Marsh [13] provided a clarification of trust concepts, presented an implementable formalism for trust, and applied a trust model to a distributed artificial intelligence (DAI) system in order to enable agents to make trust-based decisions. Marsh divided trust into three categories: **1. Basic Trust** – This is the level of trust which represents the general trust disposition of agent X ∈ 2 A at time t. 2**. General Trust** – Given agents x, y ∈ A, the general trust Tx(y)t represents the amount of trust that x has in y at time t. **3. Situational Trust** – Given agents x, y ∈ A, and a situation α, the situational trust Tx(y,α)t represents the amount of trust that x has in y in situation α at time t.

Beth et al. [14] also proposed a trust model for distributed networks. They derived trust recommendations from direct trust and gave them formal representations, as well as rules to derive trust relationships and algorithms to compute trust values. Josang et al. [15] describe a trust model where positive and negative feedback about a specific member is accumulated. The model is based on the Bayesian network model, using the beta probability density function to calculate a member's expected future behavior.

Trust is considered to be more than the authorized nature of security relations between human societies, which achieve stable and healthy operation, to a large extent thanks to the trust relationship between the individuals, groups and organizations. Therefore, in a large number of dynamic user-oriented open network environments, the study of the trust relationships between the trust-based security mechanisms to ensure the safe operation of distributed applications has become a fundamental topic. Currently, most scholars have reached a consensus that trust should have three important features [16]. **1) Subjectivity** (different entities of the same view of things which will be affected by factors such as individual preferences may vary); **2) The expected probability** (the degree of trust can be extracted and formalized as the estimated likelihood of a given event); **3) Relevance** (trust is an aspect of things, for specific content).

In recent works on trust, mainly two distinct methods are used for subjective trust reasoning: probabilistic reasoning based on statistical hypothesis testing; and approaches based on fuzzy theory, expert systems and artificial intelligence techniques. However, these methods do not fully reflect the essential nature of trust. Subjective trust, in essence, is based on the belief that it has great uncertainty. In the subjective, objective world, random and fuzzy uncertainties are the two main forms that have become the industry consensus [16]. Thus, the axiomatic methods based on probability theory or fuzzy set theory doesn't achieve a comprehensive assessment of trust information.

3.4. Trust in the Cloud

Trust and security have become crucial to guarantee the healthy development of cloud platforms, providing solutions for concerns such as the lack of privacy and protection, the guarantee of security and author rights.

Privacy and security have been shown to be two important obstacles concerning the general adoption of the cloud computing paradigm. In order to solve these problems in the IaaS service layer, a model of trustworthy cloud computing which provides a closed execution environment for the confidential execution of virtual machines was proposed [5]. This work has shown how the problem can be solved using a Trusted Platform Module. The proposed model, called Trusted Cloud Computing Platform (TCCP), is supposed to provide higher levels of reliability, availability and security. In this solution, there is a cluster node that acts as a Trusted Coordinator (TC). Other nodes in the cluster must register with the TC in order to certify and authenticate its key and measurement list. The TC keeps a list of trusted nodes. When a virtual machine is started or a migration takes place, the TC verifies whether the node is trustworthy so that the user of the virtual machine may be sure that the platform remains trustworthy. A key and a signature are used for identifying the node. In the TCCP model, the private certification authority is involved in each transaction together with the TC [5].

Shen et al. [17] presented a method for building a trustworthy cloud computing environment by integrating a Trusted Computing Platform (TCP) to the cloud computing system. The TCP is used to provide authentication, confidentiality and integrity [17]. This scheme displayed positive results for authentication, rule-based access and data protection in the cloud computing environment.

Cloud service providers (CSP) should guarantee the services they offer, without violating users' privacy and confidentiality rights. Li et al. [18] introduced a multi-tenancy trusted computing environment model (MTCEM). This model was designed for the IaaS layer with the goal of ensuring a trustworthy cloud computing environment to users. MTCEM has two hierarchical levels in the transitive trust model that supports separation of concerns between functionality and security. It has 3 identity flows: a) the consumers, who hire the CSP cloud computing services; b) the CSP, that provides the IaaS services; c) the auditor (optional, but recommended), who is responsible for verifying whether the infrastructure provided by the CSP is trustworthy on behalf of users. In MTCEM, the CSP and the users collaborate with each other to build and maintain a trustworthy cloud computing environment.

Zhimin et al. [19] propose a collaborative trust model for firewalls in cloud computing. The model has three advantages: a) it uses different security policies for different domains; b) it considers the transaction contexts, historic data of entities and their influence in the dynamic measurement of the trust value; and c) the trust model is compatible with the firewall and does not break its local control policies. A model of domain trust is employed. Trust is measured by a trust value that depends on the entity's context and historical behavior, and is not fixed. The cloud is divided in a number of autonomous domains and the trust relations among the nodes are divided in intra and inter-domain trust relations. The intra-domain trust relations are based on transactions operated inside the domain. Each node keeps two tables: a direct trust table and a recommendation list. If a node needs to calculate the trust value of another node, it first checks the direct trust table and uses that value if the value corresponding to the desired node is already available. Otherwise, if this value is not locally available, the requesting node checks the recommendation list in order to determine a node that has a direct trust table that includes the desired node. Then it checks the direct trust table of the recommended node for the trust value of the desired node. The process continues until a trust value for the desired node is found in a direct trust table of some node. The inter-domain trust values are calculated based on the transactions among the inter-domain nodes. The inter-domain trust value is a global value of the nodes direct trust values and the recommended trust value from other domains. Two tables are maintained in the Trust Agents deployed in each domain: form of Inter-domain trust relationships and the weight value table of this domain node.

In [20] a trusted cloud computing platform (TCCP) which enables IaaS providers to offer a closed box execution environment that guarantees confidential execution of guest virtual machines (VMs) is proposed. This system allows a customer to verify whether its computation will run securely, before requesting the service to launch a VM. TCCP assumes that there is a trusted coordinator hosted in a trustworthy external entity. The TCCP guarantees the confidentiality and the integrity of a user's VM, and allows a user to determine up front whether or not the IaaS enforces these properties.

The work [4] evaluates a number of trust models for distributed cloud systems and P2P networks. It also proposes a trustworthy cloud architecture (including trust delegation and reputation systems for cloud resource sites and datacenters) with guaranteed resources including datasets for on-demand services.

4. FILE DISTRIBUTION IN CLOUD

Cloud computing offers great flexibility for users, due to the fact that users don't have to worry about management complexity related to each system, for example, the databases can be transferred to data centers of large specialized companies, although the management data in outsourced environments aren't always reliable. Users are becoming dependent on the availability and integrity offered by storage service providers. Thus, it is necessary to use models of secure data storage in order to ensure the integrity of cloud users data [21].

One of the problems that cloud computing is able to solve is the storage of files and their distribution with high rate of availability. There are several approaches to manage data in the cloud and each system uses a specific approach to persist data. Among these approaches, we can highlight new file systems, frameworks and proposals for storage and processing data.

4.1. Google File System

The Google File System (GFS) is a proprietary distributed file system developed by Google and specially designed to provide efficient and reliable access to data, using large server clusters [22]. The GFS architecture consists of three elements: Clients, Master and chunkservers. A GFS cluster consists of a single master and multiple chunkservers that is accessed by multiple clients, as shown in Figure 3.

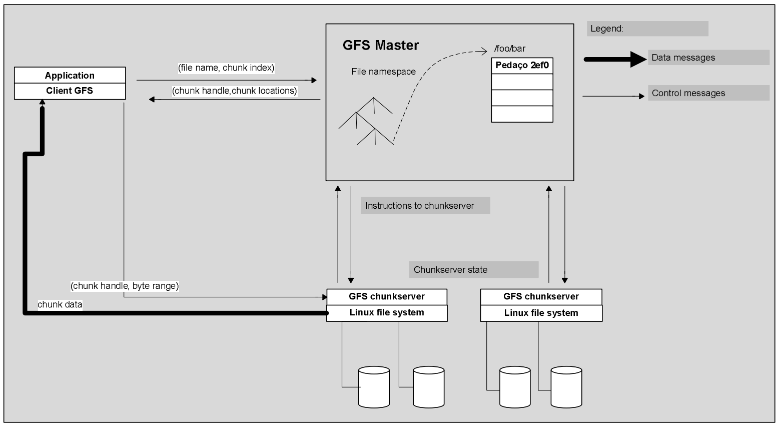

Figure 3. Architecture of GFS [22]

The master stores three major types of metadata: file and chunk namespaces, mapping from files to chunks, and the locations of each chunk's replicas. All metadata is updated by the master server, which communicates regularly with each chunkserver through the exchange of messages called heartbeat messages, to give it instructions and collect its state.

4.2. Amazon S3

The Amazon Simple Storage Service (S3) is a distributed storage system based on Dynamo [23]. Dynamo uses key-value model stored in a Distributed Hash Table (DHT) and has no support associations or schemes. To ensure a level of scalability and availability, data is partitioned and replicated in multiple machines, using a consistent hashing, being consistency facilitated by multiple versions of objects.

The consistency between replicas during updates is maintained by a quorum-like technique and a decentralized replica synchronization protocol. Dynamo employs a gossip based distributed failure detection and membership protocol. Dynamo is a completely decentralized system with minimal need for manual administration. Storage nodes can be added and removed from Dynamo without requiring any manual partitioning or redistribution.

Dynamo stores objects associated with a key through a simple interface; it exposes two operations: get() and put(). The get(key) operation locates the object replicas associated with the key in the storage system and returns a single object or a list of objects with conflicting versions along with a context. The put(key, context, object) operation determines where the replicas of the object should be placed based on the associated key, and writes the replicas to disk. The context encodes system metadata about the object that is opaque to the caller and includes information such as the version of the object. The context information is stored along with the object so that the system can verify the validity of the context object supplied in the put request.

4.3. Microsoft Azure

Microsoft SQL Azure is compound of a set of services for storing and processing data in cloud [24]. SQL Azure with Windows Azure Storage compose the solution of data management in cloud of Microsoft. The purpose of Windows Azure Storage is provide a scalable storage, durable, highly available and provide users the payment on demand. It allows easy access to data, providing a simple interface, remotely available and in data centers. The storage services in Windows Azure Storage are offered in four levels of abstraction: blobs, table, drives and queue structures. Windows Azure Storage includes persistent storage through blobs, tables and queues. The storage access and load balancing is done automatically through a set of nodes responsible for physical storage providing scalability and availability.

To use Windows Azure Storage service, user needs to create a storage account, which can be obtained from the Windows Azure portal web interface. After the creation of an account, user will receive a 256-bit secret key.

In Azure data storage Microsoft promises to keep the confidentiality of users' data. The procedure shown in Figure 4 provides security for data accessing to ensure that data will not be lost.

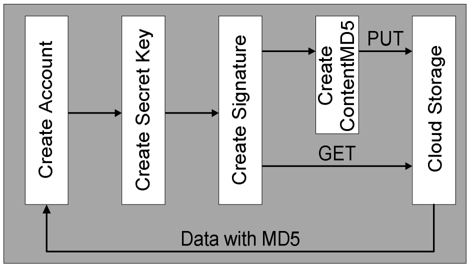

Figure 4. Security data access procedure [25]

4.4. Hadoop

Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware [26] and its objective is storing large amounts of data across multiple nodes.

HDFS has master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode. The Figure 5 shown architecture distributed file system HDFS.

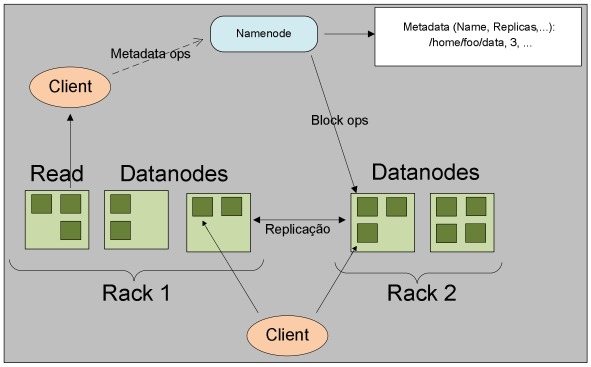

Figure 5. Architecture Distributed File System HDFS [26]

Hadoop store large files on multiple servers and get the reliability through data replication. Similar to the GFS, the data are stored in geographically distributed nodes.

4.5. Secure Distributed Data Storage in Cloud Computing

One of the core services provided by cloud computing is data storage. It poses new challenges in creating secure and reliable data storage and access facilities over remote service providers in the cloud. The security of data storage is one of the necessary tasks to be treated before the project for cloud computing be fully accepted.

There are at least two concerns when using the cloud. One concern is: users do not want to reveal their data to the cloud service provider. Another concern is that users are unsure about the data integrity that they receive from the cloud. Therefore, within the cloud, more than conventional security mechanisms will be required for data security. One of the main challenges that avoid end users from adopting cloud service storage is the fear of losing data or violation of it. Thus, the data integrity and unreliable storage is a major challenge for providers of cloud storage [28]. It is essential to minimize the fear of users, provide technologies that able user to verify the integrity of your data.

Service providers of cloud file storage presented don’t solve problems related to the reliable exchange of files between peers. The same concern is related to the availability of files, solving this problem through file replica.

Encryption techniques can be used to ensure data privacy in the cloud. However, these techniques have significant performance implications of queries in SGBDs. Thus, alternatives for integrating encryption techniques with database management systems should be evaluated and proposed, since data encryption computational complexity increases the response time of the query. In addition, it is necessary to propose a model for a reliable files exchange in the cloud computing environment, because this problem has not been solved.

5. HIGH LEVEL TRUST MODEL FOR RELIABLE FILE SHARING

According to the review and related research [5] [11] [17] [19] [21] [27], it is necessary to employ a cloud computing trust model to ensure the exchange of files among cloud users in a trustworthy manner. In this section, we introduce a trust model to establish a ranking of trustworthy nodes and enable the secure sharing of files among peers in a private cloud. The environment computing private cloud was chosen because we work with a specific context of distributing files, where the files have a desired distribution and availability.

We propose a trust model where the selection and trust value evaluation that determines whether a node is trustworthy can be performed based on node storage space, operating system, link and processing capacity. For example, if a given client has access to a storage space in a private cloud, it still has no selection criterion to determine to which cloud node it will send a particular file. When a node wants to share files with other users, it will select trusted nodes to store this file through the proposed following metrics: processing capacity (the average workload processed by the node, for example, if the node's processing capacity is 100% utilized, it will take longer to attend any demands), operating system (operating system that has a history of lower vulnerability will be less susceptible to crashes), storage capacity and link (better communication links and storage resources imply greater trust values, since they increase the node's capacity of transmitting and receiving information). The trust value is established based on queries sent to nodes in the cloud, considering the metrics previously described.

Each node maintains two trust tables: direct trust table and the recommended list. a) If a node needs to calculate the trust value of another node, it first checks the direct trust table and uses the trust value if the value for the node exists. If this value is not available yet, then the recommended lists are checked to find a node that has a direct trust relationship with the desired node the direct trust value from this node's direct trust table is used. If there's no value attached, then it sends a query to its peers requesting information on their storage space, processing capacity and link. The trust values are calculated based on queries exchanged between nodes.

b) The requesting node will assign a greater trust value to nodes having greater storage capacity and / or processing and better link. In addition, the operating system will also be considered as a criterion of trust.

Figure 6 presents a high level view the proposed trust model, where the nodes query their peers to obtain the information needed to build their local trust table.

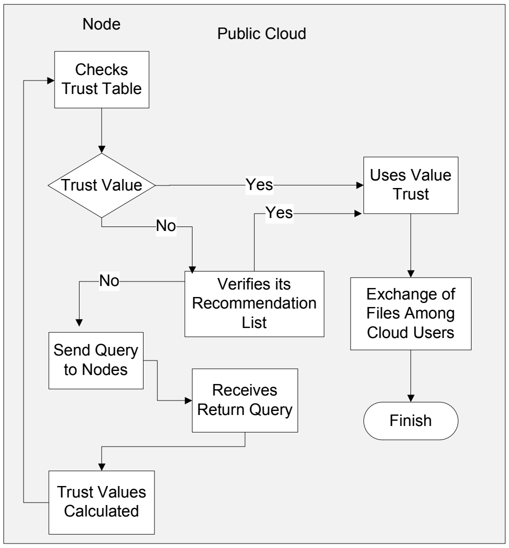

Figure 6. High Level Trust Model

In this model, a trust rank is established, allowing a node A to determine whether it is possible to trust a node B to perform storage operations in a private cloud. In order to determine the trust value of B, node A first has to obtain basic information on this node. The scenario of information request for a reliable file exchange between nodes is presented Figure 7.

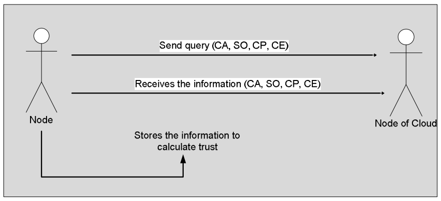
Send query (CA, SO, CP, CE)
Receives the information (CA, SO, CP, CE)
Node
Node of Cloud
Stores the information to calculate trust

Figure 7. Scenario of Information Request

When node A needs to exchange a file in cloud and it wants to know if node B is trusted to send and store the file, it will use the proposed Protocol Trust Model, which can be described with the following scenario: step 1, node A sends a request to the nodes of cloud, including node B, asking about storage capacity, operating system, processing capacity and link. In step 2, nodes, including node B, send a response providing the requested information. In step 3, node A evaluates the information received from B and from all nodes. If the information provided by B, are consistent with the expected, with the average value of the information of other nodes, the values are stored in local recommendations table of node A, after to make the calculation of trust and store in your local trust table. The trust value of a node indicates its disposition/suitability to perform the operations between cloud peers. This value is calculated based on the interactions/queries historical between nodes, having value ranging between [0, 1].

In general, trust of node A in node B, in the context of a private cloud NP, can be represented by a value V which measures the expectation that a particular node will have good behaviour in the private cloud, so trust can be expressed by:

The trust value of a node indicates its disposition/suitability to perform the operations between peers of cloud. This value is calculated based on the history interactions/queries between the nodes, can to have value ranging between [0, 1].

In general, the trust of node A in node B, in the context of a private cloud NP, can be represented by a value V which measures the expectation that a particular node will behave in the private cloud, so the trust can be expressed by:

$$\mathrm{T}_{(\mathrm{a,b})}^{\mathrm{np}} = \mathrm{V}_{\mathrm{np}}^{\mathrm{b}} \quad (1)$$

$T_{(a,b)}^{np}$ Represent the trust of A in B in the private cloud NP and $\mathrm{V}_{\mathrm{np}}^{\mathrm{b}}$ represent of value trust of B, in the private cloud NP analyzed by A.

According to definition of trust, $\mathrm{V}_{\mathrm{np}}^{\mathrm{b}}$ is equivalent to queries sent and received (interaction) by A related to B in cloud NP. As the interactions are made between the nodes of private cloud, the information is used for the calculation of trust.

Nodes of a private cloud should be able to consider whether a trust value is acceptable, generating trust level. If the node exceeds the level within a set of analyzed values, it must be able to judge the node in a certain degree of trust. Trust degree can vary according to a quantitative evaluation: a node has a very high trust in another one, a node has low trust in another one, a node doesn’t have sufficient criteria to opine, a node trusts enough to opine, etc.

Table 2 represents values that were established to determine the trust and reputation quantitative evaluation in a node.

Table 2. Reference values for trust

Value	Description	Decision
0	No trust in nodes in the private cloud	No opinion
[0, 0.39]	Low trust in nodes in the private cloud	No trust
[0.4, 0.59[	Average trust in nodes in the private cloud	No trust
[0.6, 0.89[	High trust in nodes in the private cloud	Trust
[0.9, 0.99[	Very High trust in nodes in the private cloud	Trust

According to information in the reference table, one node trusts another node from trust value T ≥ 0.6. The trust values are calculated from queries between the nodes of NP, allowing to obtain the necessary information for final calculation of trust. The trust information is stored through the individual records of interaction with the respective node, staying in local database information about the behavior of each node in the cloud that wants to exchange a file (local trust table and local recommendations table). The calculation of trust of a node A in B in cloud NP will be represented by [28] [29]:

$$T_{(a,b)}^{fnp} = \frac{\sum_{np=1}^{j} V_{np}^{b}\,(b, m_1) + (b, m_2) + (b, m_3) + \cdots + (b, m_n) \le 1}{\mathrm{j}} \qquad (2)$$

$\mathrm{T}_{(\mathrm{a,b})}^{\mathrm{fnp}}$ Represents the final trust of A in B in cloud NP. The trust value of B is defined as the sum of metrics values that the node B has (m) in the cloud NP; j represents the number of interactions of trust from node A in B in the cloud NP, where j ≥ 0.

Four aspects can to have impact on calculation of direct trust of a node, as shown in Table 3. Greater storage capacity and processing capacity have more weight in the choice of a node more reliable, because of these features are the responsible for ensure the integrity and file storage. To calculate direct trust of a node, it is attributed by administrator of the private cloud: storage capacity and processing with weights of 35%, 15% to link and the remaining 15% to operating system. Knowing that a node can to have the trust value ranging from [0.1] and that these values are variable over time, a node can have its storage capacity increased or decreased, it's necessary that trust reflects the behavior of a node in a given period of time. Nodes with constant characteristics should therefore be more reliable because they have less variation in basic characteristics.

Table 3 - Aspects Affecting the Trust Direct Node

Storage capacity	Processing capacity	Link capacity	operating system	Trust Direct
High	High	High	High	High
High	High	Low	Low	High
High	Low	High	Low	Average (It depends on the values Storage and Processing)
High	Low	Low	Low	Low
Low	High	High	Low	Average (It depends on the values Storage and Processing)
Low	High	Low	Low	Low
Low	Low	High	Low	Low
Low	Low	Low	Low	Low

According to the weights attributed it's possible to calculate the trust of node using this formula:

$$T_{(a,b)}^{fnp} = \frac{\sum_{np=1}^{j} V_{np}^{b} \left((b,m_1) * 0{,}35\right) + \left((b,m_2) * 0{,}35\right) + \left((b,m_3) * 0{,}15\right) + \left((b,m_n) * 0{,}15\right) \leq 1}{j} \quad (3)$$

5.1. Initial Results and Simulations

Once assigned the weights metrics, can perform the calculation of the trust of a node. Consider the node A and B and between them execute 10 iterations (j). The simulation is started by performing the calculation with the node A trusting B, is assigned the value 1 to all metrics.

To perform the simulation we used the Monte Carlo method [31] for the generation of random numbers or pseudo-random, for four metrics: Storage Capacity, Processing Capacity, Operating System and Link. Thus, from the first iteration, the values of each of these metrics are assigned randomly varying between 0 and 1, as shown in Table 4.

Table 4 -Simulation Calculation of Trust

Initial Iteration						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
1	1	1	1	1	1	Trust
Iteration 2						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,01	0,82	0,57	0,91	0,51	0,76	Trust
Iteration 3						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,35	0,08	0,62	0,89	0,38	0,63	Trust
Iteration 4						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,28	0,96	0,85	0,14	0,58	0,62	Trust
Iteration 5						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,06	0,11	0,30	0,76	0,22	0,54	Not Trust
Iteration 6						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,68	0,01	0,50	0,89	0,45	0,52	Not Trust
Iteration 7						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,15	0,33	0,16	0,48	0,26	0,49	Not Trust
Iteration 8						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,70	0,34	0,61	0,79	0,57	0,50	Not Trust
Iteration 9						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,25	0,95	0,53	0,89	0,63	0,51	Not Trust

Iteration 10						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,58	0,80	0,72	0,64	0,69	0,53	Not Trust
Iteration 11						
CP (35%)	CA(35%)	CE(15%)	SO(15%)	CD	CDFinal	Decision Trust
0,27	0,54	0,95	0,19	0,46	0,52	Not Trust

Observing Table 4 can see that the values of the metrics in each simulation directly influence the decision to trust or not in node.

6. CONCLUSIONS

Cloud computing has been focus of several recent research, which demonstrates the importance and need of trust model that ensures reliable and secure exchange files.

We have presented an overview of the cloud computing paradigm, as well as its main features, architectures and deployment models. Moreover, we identified the main issues related to trust, privacy and security in cloud computing environments.

In order to address these issues, we proposed a trust model to ensure reliable exchange of files among users in a private cloud environment, using the concepts of trust and reputation, that has been promissory due to identification of problems and vulnerabilities related to security, privacy and trust that a cloud computing environment presents. In our model, the trust value of a given node is obtained from a pool of simple parameters related to its suitability for performing storage operations. Nodes with greater trust values are subsequently chosen for further file storage operations.

As a future work, we plan to implement the proposed trust model and analyze node behavior after the ranking of trustworthy nodes is established.

REFERENCES

[1] Minqi Zhou, Rong Zhang, Dadan Zeng, and Weining Qian, “Services in the cloud computing era: a survey,” Software Engineering Institute. Universal Communication. Symposium (IUCS), 4th International. IEEE Shanghai, pp. 40-46. China. 978-1-4244-7821-7 (2010).

[2] Xue Jing and Zhang Jian-jun, “A Brief Survey on the Security Model of Cloud Computing,” 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), Hong Kong IEEE, pp. 475 – 478. Aug 2010.

[3] P. Mell and T. Grance, "Draft nist working definition of cloud computing - v15," 21. Aug 2009.

[4] T. Dillon, Chen Wu, and E. Chang, “Cloud Computing: Issues and Challenges,” 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 27-33. Australia, 2010.

[5] Xiao-Yong Li, Li-Tao Zhou, Yong Shi, and Yu Guo, “A Trusted Computing Environment Model in Cloud Architecture,” Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, 978-1-4244-6526-2. Qingdao, pp. 11-14. China. July 2010.

[6] A. Marinos and G. Briscoe, “Community cloud computing,” in First International Conference Cloud Computing, CloudCom, volume 5931 of Lecture Notes in Computer Science, pp. 472–484. Springer (2009).

[7] H. Takabi, J. B. D. Joshi, and G. Ahn, “Security and Privacy Challenges in Cloud Computing Environments,” IEEE Security and Privacy, vol. 8, no. 6, pp. 24-31, Nov./Dec. 2010, doi:10.1109/MSP.2010.186.

[8] Zhidong Shen, Li Li, Fei Yan, and Xiaoping Wu, “Cloud Computing System Based on Trusted Computing Platform,” Intelligent Computation Technology and Automation (ICICTA), IEEE International Conference on Volume: 1, pp. 942-945. China. 2010.

[9] Huan-Chung Li, Po-Huei Liang, Jiann-Min Yang, and Shiang-Jiun Chen, “Analysis on Cloud-Based Security Vulnerability Assessment,” 2010 IEEE 7th International Conference on e-Business Engineering (ICEBE), pp. 490-494, 2010.

[10] K. Popovic and Z. Hocenski, “Cloud computing security issues and challenges,” MIPRO, 2010 Proceedings of the 33rd International Convention, pp. 344-349, 24-28 May 2010 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5533317&isnumber=5533310.

[11] S. Uppoor, M. Flouris, and A. Bilas, “Cloud-based synchronization of distributed file system hierarchies,” Cluster Computing Workshops and Posters (CLUSTER WORKSHOPS), IEEE International Conference, pp. 1-4. 2010.

[12] Qingsong Wei, Bharadwaj Veeravalli, Bozhao Gong, Lingfang Zeng, and Dan Feng, “CDRM: A Cost-Effective Dynamic Replication Management Scheme for Cloud Storage Cluster,” 2009 IEEE International Conference on Cluster Computing (CLUSTER), pp. 188-196, 2010.

[13] S. P. Marsh, “Formalising Trust as a Computational Concept”, Ph.D. Thesis, University of Stirling, 1994.

[14] T. Beth, M. Borcherding, and B. Klein, “Valuation of trust in open networks,” In ESORICS 94. Brighton, UK, November 1994.

[15] A. Jøsang and R. Ismail, “The Beta Reputation System,” In Proceedings of the 15th Bled Electronic Commerce Conference, pp. 17-19. June 2002.

[16] A. Abdul-Rahman and S. Hailes, “A distributed trust model,” In Proceedings of the 1997 New Security Paradigms Workshop, pp. 48-60, 1998.

[17] Zhidong Shen, Li Li, Fei Yan, and Xiaoping Wu, “Cloud Computing System Based on Trusted Computing Platform,” Intelligent Computation Technology and Automation (ICICTA), IEEE International Conference on Volume: 1, pp. 942-945. China. 2010.

[18] Li Xiaoqi, Lyu M R, and Liu Jiangchuan. “A trust model based routing protocol for secure AD Hoc network,” Proceedings of the 2004 IEEE Aerospace Conference, pp. 1286-1295. 2004.

[19] Zhimin Yang, Lixiang Qiao, Chang Liu, Chi Yang, and Guangming Wan, “A collaborative trust model of firewall-through based on Cloud Computing,” Proceedings of the 2010 14th International Conference on Computer Supported Cooperative Work in Design. Shanghai, China. pp. 329-334, 14-16. 2010.

[20] N. Santos, K. Gummadi, and R. Rodrigues, “Towards Trusted Cloud Computing,” Proc. HotCloud. June 2009.

[21] Wang, J., Shao, Y., Jiang, S., e Le, J. “Providing privacy preserving in cloud computing”. Em International Conference on Test and Measurement, páginas 213–216. IEEE Computer Society. Hong Kong, 2009.

[22] Ghemawat, S., Gobioff, H., and Leung, Shun-Tak. The google file system. Proceedings of the nineteenth ACM symposium on Operating systems principles ACM. New York, Volume 37 Issue 5, December 2003. NY, USA.

[23] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels. “Dynamo: amazon’s highly available key-value store”. Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles. ACM. New York, NY, USA. 2007.

[24] Azure. Microsoft Azure. 2011. http://www.microsoft.com/azure/

[25] Rajkumar Buyya, James Broberg, Andrzej Gościński. “Cloud computing: principles and paradigms”. Hoboken, N.J. Wiley, 2011.

[26] D. Borthakur, "The Hadoop Distributed File System: Architecture and Design". The Apache Software Foundation, 2007. http://hadoop.apache.org accessed 06/12/2011.

[27] Kai Hwang, Sameer Kulkareni, and Yue Hu, "Cloud Security with Virtualized Defense and Reputation-Based Trust Mangement," 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC '09), pp. 717-722, 2009.

[28] Canedo, Edna Dias, Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Junior. Review of Trust-based File Sharing in Cloud Computing. In: The Fourth International Conference on Advances in Mesh Networks - MESH 2011 - August 21-27, 2011 - French Riviera, 2011, Nice/St Laurent du Var - France. IARIA Conference - MESH 2011, The Fourth International Conference on Advances in Mesh Networks. , 2011. p.44 – 50.

[29] Canedo, Edna Dias, Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Junior. Trust Model for File Sharing in Cloud Computing. In: CLOUD COMPUTING 2011: The Second International Conference on Cloud Computing, GRIDs, and Virtualization, 2011, Rome - Italy. IARIA Conference - CLOUD COMPUTING, International Conference on Cloud Computing, GRIDs, and Virtualization. , 2011. p.66 – 73.

[30] Kathleen Jungck1 and Syed (Shawon) M. Rahman. CLOUD COMPUTING AVOIDS DOWNFALL OF APPLICATION SERVICE PROVIDERS. International Journal of Information Technology Convergence and Services (IJITCS) Vol.1, No.3, June 2011. Minneapolis, USA. 2011.

[31] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery. Numerical Recipes: The Art of Scientific Computing, Third Edition (2007), published by Cambridge University Press (ISBN-10: 0521880688, or ISBN-13: 978-0521880688).

Authors

Edna Dias Canedo is graduated in Systems Analysis by Universidade Salgado de Oliveira; Goiás (1999). Master degree by Universidade Federal da Paraíba UFPB in the domain of software engineering (2002). She is currently following her PhD program at Universidade de Brasília (UNB) and is professor of the Software Engineer Course in the Gama Institute, of the Universidade de Brasília – UNB.

Rafael Timóteo de Sousa Jr. is graduated in Electrical Engineering by Universidade Federal da Paraíba, Campina Grande (Brazil, 1984), Master degree (DEA) in Telematics and Information Systems by École Supérieure d'Electricité - SUPELEC (France, 1985) and Doctorate degree in Signal Processing and Telecommunications by Université de Rennes I (France, 1988). He performed his sabbatical year research on computational trust in ad hoc networks at Ecole Supérieure d'Electricité - SUPELEC (2006-2007). He is an associate professor of the Universidade de Brasília in the domains of Computer Networks Engineering, Information Technology and Information Security.

Robson de Oliveira Albuquerque is graduated in Computer Science by Universidade Católica de Brasília (1999). Specialization in Computer network by União Educacional de Brasília (2000). master's degree by Universidade de Brasília (2003). Master degree (DEA) in computer systems and programming by Universidade Complutense de Madri (2007). Doctorate in Electrical Engineering by Universidade de Brasília (2008). Presently he is a federal public employee, researcher at Department of Electrical Engineering of UnB, researcher associated to the Grupo de Análisis, Seguridad y Sistemas (GASS) at Universidade Complutense de Madri and he is following a doctorate degree program at Universidade Complutense de Madri.