Exploring a New Security Framework for Cloud Storage Using Capabilities

Glenford Mapp and Mahdi Aiash
School of Science
and Technology
Middlesex University,
Hendon, London NW4 4BT
Email: g.mapp, m.aiash@mdx.ac.uk

Brian Ondiege and Malcolm Clarke
School of Information Sciences,
Computing and Mathematics
Brunel University,
Kingston Lane, Uxbridge
Middlesex UB3 3PH
Email: brian.ondiege, malcolm.clarke@brunel.ac.uk

***Abstract*—We are seeing the deployment of new types of networks such as sensor networks for environmental and infrastructural monitoring, social networks such as facebook, and e-Health networks for patient monitoring. These networks are producing large amounts of data that need to be stored, processed and analysed. Cloud technology is being used to meet these challenges. However, a key issue is how to provide security for data stored in the Cloud. This paper addresses this issue in two ways. It first proposes a new security framework for Cloud security which deals with all the major system entities. Secondly, it introduces a Capability_ID system based on modified IPv6 addressing which can be used to implement a security framework for Cloud storage. The paper then shows how these techniques are being used to build an e-Health system for patient monitoring.**

***Index Terms*—Cloud Storage, Security Framework, Capability Systems, e-Health Monitoring**

I. Introduction

The term Big Data encapsulates the large amounts of data being produced in digital environments. New networks, such as sensor networks, e-Health systems for patient monitoring [1] and social networks which promote human interaction, are producing data at a phenomenal rate. This data needs to be efficiently stored, processed and analysed. Cloud technology, being developed by Cloud Providers such as Amazon Services and Rackspace, is being deployed to meet these challenges.

However, a major issue remains of how well data can be secured in a Cloud environment. Data needs to be accessible not just to producers and consumers of that data but also to Cloud Infrastructure Agents who might need to migrate or replicate data based on access demands. For example, moving data, beforehand, nearer to the compute engines that will process the data may bring significant energy savings. Security mechanisms must therefore take account of such realities.

Though there have been several research efforts trying to deal with the security of Cloud storage [2], the authors believe a new and comprehensive framework is required to provide the kind of operational flexibility needed in Cloud environments. Furthermore, the issue of how the proposed functions in this Cloud Security framework are developed into mechanisms needs to be addressed. The authors believe that a capability-based approach could be adopted with great effect to develop the required mechanisms. Hence this paper explores the development of a Capability_ID system and shows how such a system could be used to build an e-Health system for monitoring patients.

The rest of this paper is structured as follows: Section 2 looks at common elements of the Cloud Environment while Section 3 details the New Security Framework for Cloud Storage. Section 4 introduces the idea of Capabilities. Section 5 introduces a new Capability_ID system while Section 6 shows how this new Capability_ID system is used to develop a Cloud Storage system for e-Health monitoring of patients. Section 7 details current work being done to build a real system. The paper concludes in Section 8.

II. Understanding Cloud Environments

The Cloud environment is a relatively new computing environment. It uses virtualization to deliver resources such as computing, memory, storage and networking facilities to applications. This is done by the use of a hypervisor which provides a virtual environment that interacts with the Cloud Infrastructure. The hypervisor therefore is the boundary between the application and the Cloud Infrastructure. Normal virtualization, in which

there is a Host OS as well as a Guest OS, has given way to Bare Metal Virtualization using para-virtualization techniques. In Bare-Metal virtualization, the hypervisor replaces the Host Operating System and directly controls the hardware.

In addition, there are three Cloud paradigms which have taken hold. IaaS, Infrastructure as a Service, works to replace company infrastructure such as desktops and servers with Cloud infrastructure using virtual machines. Amazon Services is an example of an IaaS company. PaaS, Platform as a Service, allows programmers that develop Cloud programs using Cloud Infrastructure via a Cloud-based Software Development Kit (SDK). The Google Application Engine (GAE) is an example of PaaS. Finally SaaS, Software as a Service, is used to provide company software such as Customer Relations Management (CRM) and Enterprise Resource Planning (ERP). Salesforce.com is an example of an SaaS provider.

A. Issues in Cloud Security

The paradigms described above do not directly address the issues of providing secure data access and storage in the Cloud because the Cloud was initially viewed as a replacement of the company's infrastructure and so the security mechanisms that were associated with servers were thought to be adequate. This has not proven to be the case for a number of reasons. Firstly there is now a recognition of the growing importance of data in the context of company identity. The data generated by a company is now regarded as a key part of a company's DNA. In addition, company data is administered in a very sophisticated way: not all data is available to everyone. Some types of data are very sensitive even within companies.

B. The Firesmith Framework for Reusable Security

The Firesmith Framework is a detailed specification [3] which attempts to provide a comprehensive security framework. It consists of nine layers: access control, attack harm detection, non-repudiation, integrity, security auditing, physical protection, privacy and confidentiality, recovery and prosecution. This framework provides a detailed analysis of the required functionality and therefore is able to serve as a reference model. Although this framework is significant, the authors believe that a new framework is required which incorporates the functionality of the Firesmith Framework in the contexts of major entities such as applications and users as well as Cloud Infrastructure facilities such as the hypervisor and Storage Servers.

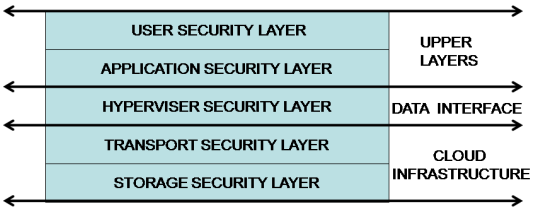

Fig. 1: The Cloud Security Framework

III. The New Security Framework

The new Security Framework is shown in Figure 1.

- User Security Layer: This layer is concerned with user authentication and authorization. Users start by authenticating themselves to the local device [4] as well as to the application. This authentication leads to the authorization of the user to use application resources.
- Application Security Layer: This layer is used to authenticate the application to the hypervisor. This layer is also used to authenticate the user to the hypervisor and vice-versa if access to the user's data is required. This security layer is also responsible for Presentation Security which encodes and decodes data between the application and the Cloud Storage System. Where the data belongs to the user, this layer acquires the relevant keys to ensure that the user's data is also protected.
- Hypervisor Security Layer: This is the layer that is implemented in the hypervisor. This layer is used to authenticate the application and user security layers to the Cloud Infrastructure. This layer is also used to generate capabilities which allow applications to access the required resources in the Cloud Infrastructure. It also checks that there is no violation due to cloud personnel accessing relevant information in order to maintain a Cloud Infrastructure. For example, the blocks of a file maybe copied by a system administrator but contents of the file should not be read by those working for the Cloud Provider. This layer therefore acts as the data interface between

the upper layers described above and the Cloud Infrastructure.

- Transport Security Layer. This layer can provide secure communication between the application and the Cloud Infrastructure. It is now possible to look at secure transport protocols which provide protection of data using secure links. New protocols such Simple Protocol (SP) [5] allow quick authentication using key exchange mechanisms such that a secure key is generated and used for moving data between the application and the Cloud Infrastructure.
- Storage Security Layer. This layer is about securing blocks of data in the Cloud Infrastructure. This will involve securing blocks using encryption techniques such as AES and DES algorithms. In addition, each block is hashed after it has been modified to ensure no unauthorised modifications. In order to ensure 100% availability, blocks may be replicated throughout the Cloud Storage structure. Therefore a coherency protocol within the storage layer is used to synchronize different copies of the block.

IV. The Use of Capabilities

Capabilities were first used in the memory protection system of the CAP computer [6] developed by the Computer Laboratory at the University of Cambridge, UK. The CAP system used a Capability Unit (CU) in which computation could only proceed if the right capability was loaded into the CU. This therefore protected different memory segments and thus facilitated multiprogramming. This idea did not catch on and the next major use of capabilities was within the Ameoba Distributed Operating System [7]. Here a capability was a software construct which specified the right of its holder to invoke operations on a defined operating system object via a communications port.

A. Using IPv6 format

IPv6 is the new generation of the Internet Protocol Suite. The IPv6 format is given in Figure 2. The development of IPv6 encouraged new approaches to address some key issues. This is because an IP address attempts do two things: it is used to identify the network interface to which the packet is sent and it is also used to route packets to the destination. This dual functionality meant that it was difficult to dynamically optimize network routes in mobile environments. This led to better support for addressing in IPv6 [8]. In addition, IPSec [9] which is a security framework for IP that allows the setting up of secure connections using encryption and tunnelling

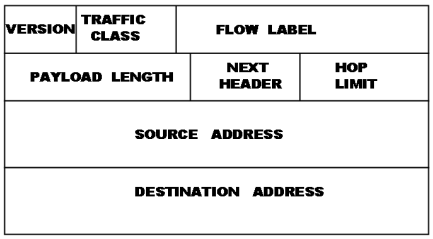

Fig. 2: IPv6 Header Format

has been designed to work with IPv6. However, though IPv6 can be regarded as a technical improvement on IPv4, these improvements have not been able to be fully used in environments such as mobile and Cloud systems where greater flexibility is required. This has led to the development of new architectures such as Y-Comm [10] and Mobile Ethernet [11] for building future mobile networks. This paper in turn suggests that IPv6 should be modified to give more support to Cloud Environments.

V. Modifying the IPv6 Address to support a Capability-ID system

The address space of IPv6 affords the opportunity to design a capability_ID based system for users, applications and Cloud infrastructure. Objects and their properties are identified by the use of capabilities. Capabilities therefore need to be carefully managed and need to be protected against being created or changed in an inappropriate manner. The new capability uses the whole 128 bit field of an IPv6 address and is shown in Figure 3.

- The Type Field (8 bits). This field is used to specify the type of object capability that is being used. Types could include Cloud Providers, Cloud platforms, users, applications, etc. In order to help administer the system, a special object type known as a **Capability List** or CL has been created. The CL is used to group a list of capabilities together. In terms of Capabilities List, there is a property called *common*. A common Capability List belongs to all the users in the system.
- The Property Field (12 bits). This field is used to define the properties of the object. This field is divided into properties to do with the capability itself and properties of the object that the capability

Fig. 3: New Capability Format

represents. The capability related fields are given by three bits:

1) The System or S bit. This indicates whether the object involved has been created by the system or by an application or user. System capabilities cannot be modified or deleted by users or applications.
2) The Master or M bit. This bit indicates that the capability was created by a Certificate Authority or CA. If this bit is zero, it means that this is a Proxy capability. Proxy capabilities are derived from Master capabilities. Proxy capabilities cannot be derived from other proxy capabilities.
3) The Change or C bit. This bit is used to indicate whether or not this capability can be changed. This means that if this bit is not set, no proxy certificates can be derived from this capability.

The other 9 bits are used to define properties of the object that is related to the type.

- The Object Id (72 bits). This field is used to uniquely identify the object. In order to be compatible with the current Internet, a location/ID split architecture [12] is used. So the first 32 bits are an IPv4 address which is used to indicate where the object is located or resides. This will apply to all the network-based entities in the Cloud Infrastructure.
- The Random Bit Field (24 bits). This field helps to uniquely identify the object. The random bit field is generated when the object is created. When Proxy certificates are created a new random field is generated.
- The Hash Field (12 bits). The Hash field is used to prevent the casual tampering of capabilities.

So when an object capability is created the type, properties and Object Id fields are first generated. Then the random bits are generated. Finally these fields are used to generate a SHA-1 hash which is placed in the Hash Field of the capability. Once a Master capability is created, if the CHANGE bit is set, then it is possible to narrow the use of the object by other entities by creating proxy capabilities.

VI. Building a System to Manage E-Health

In this section we look at how we would use the Capability_ID system and the Security Framework in a system to monitor patients. This system has been designed to allow patients to be monitored from their homes.

Firstly, we start by identifying key entities in the system. We detail these entities below:

- Users: There are different groups of users of the e-Health system including patients, carers, nurses, doctors, consultant. This arrangement fits well into a role-based security scheme in which each user is given a unique capability and a role-based property which specifies their role within the Health System. Each health professional will have four Capability Lists: the first is their private CL which is known only to them. The second CL is a common CL for their role. So there is a common CL for doctors, another for nurses, etc. The third CL is a common CL for the Health Profession as a whole. So things such as codes of conduct, procurement procedures, and other professional practices are handled by this CL. The four CL is a common CL that is used by everyone working on a particular site.
- Devices: There are a number of devices also involved. The first is an e-Health band which is worn by the patient. The band itself has its own object_ID but it also stores the object_ID of the patient as well as the object_ID of the server. The monitoring device uses the local Wi-Fi network to connect to the Server at the surgery/hospital which is responsible the patient.
- Servers: Servers are machines through which patients records are accessed. The servers run e-Health monitoring applications which use a special Distributed File System to store patients' records. The Distributed File System uses the Cloud to store blocks of data.
- Cloud Block Storage Systems: The Block Servers can retrieve blocks that have been identified as part

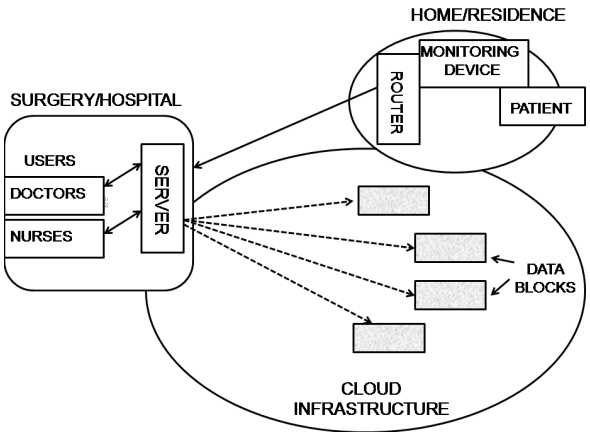

Fig. 4: The Remote Patient Monitoring System

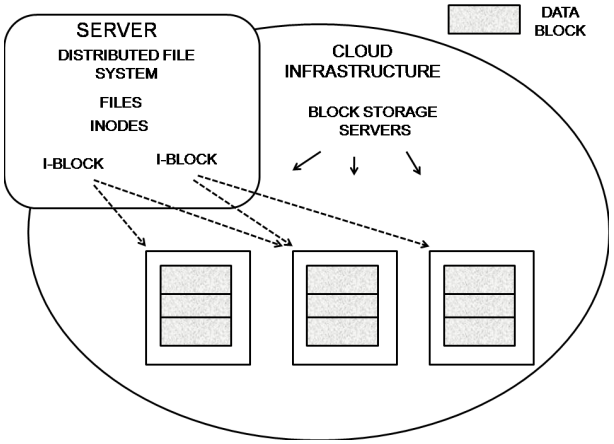

Fig. 5: New File Structure

of a file. Blocks can be replicated and migrated based on load demand and user mobility.

A. Operational Issues

When the monitoring device needs to take a reading, it reads the band and finds the server object to which the data must be forwarded. The monitoring device makes a secure connection to the server using a PKE mechanism to exchange a shared key. When the server is given the latest reading, it reads the object_ID of the patient and generates a new file in which the latest readings are stored. After the data is stored in the Cloud, a notification is sent to the doctor who is looking after the patient. The notification has the name of the file which contains the new readings. The data blocks of this new file are stored in the Cloud. The operational diagram is shown in Figure 4.

B. The File System

The File System is a Distributed File System that encrypts all the data blocks. The data blocks are replicated and placed randomly on a number of Cloud Block Storage servers. In order to improve security the inode or meta-data part of the file is not stored in the Cloud. The meta-data is protected so that if an intruder manages to decode a data block; it would still be very difficult to read the whole file. The new file system is shown in Figure 5

VII. Current Work

A. Secure Transport in LAN/Cloud Environments

In this section, we look at the need to develop secure transport which can also perform efficiently in the LAN/Cloud Environment. An examination of network interactions at the local area level clearly indicates that there is a need for much more transactional support at the LAN/Cloud level as there is a large amount of client/server interaction in order to use network services. Some services such as NFS have developed their own reliable transport [13], but there is a need for a more generic protocol of this type which can be used by many client/server applications. In addition, LAN speeds are continuing to increase in both wired and wireless networks. In wired networks, 1 Gbps is quite common and 10 Gbps is already being deployed [14]. There have been several attempts to tune WAN protocols like TCP, but these has been met by limited success.

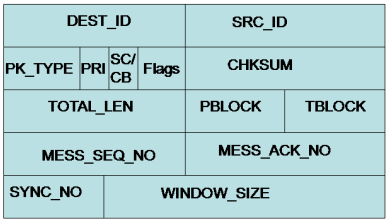

Fig. 6: The Simple Protocol

1) The Simple Protocol: In order to explore these issues further, a new transport protocol has been developed which is optimised for the local area. Called the Simple Protocol or SP, its specification is detailed in [5] and

shown in Figure 6. Unlike TCP, SP is a message-based protocol, where messages are divided into blocks before transmission over the network interface. SP provides a reliable service and can run over unreliable data substrates such as UDP or raw Ethernet. A secure version of SP has been specified and has been tested using AVISPA [15].

B. Preliminary Results

In order to implement a totally secure environment, we need to have transport mechanisms which are more closely controlled by the application or the server. This means that the transport protocol must be able to run both in User Space (US) and Kernel Space (KS). Clients run the protocol in User Space, but servers may run in User Space or Kernel Space depending on the complexity of the service. However, we need to know if running in User Space imposes an unacceptable degradation in performance. So we have conducted preliminary tests which measured the situations of both client and server are running in User Space; and where the client is in User Space and the server is Kernel Space. These results were compared with a standard in-kernel version of TCP. These tests were performed in a Network Laboratory on a 1 Gbps Ethernet Network with no effective load. SP ran over UDP and the transport data block size was kept to 1472 bytes so as to mimic segmentation done by TCP. The machines ran up-to-date versions of the Linux kernel. The results are shown in Figure 7. This shows that SP can perform effectively in User Space and Kernel Space while providing greater flexibility.

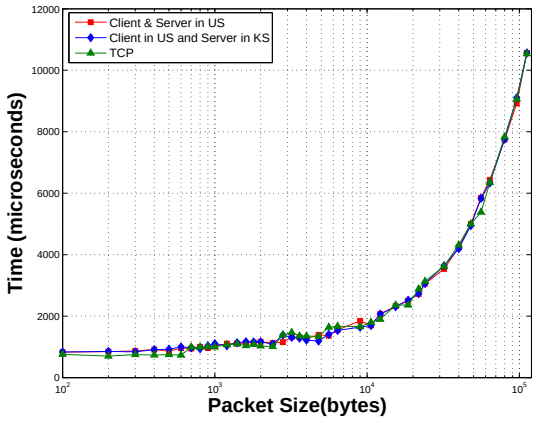

Fig. 7: Preliminary Performance of SP

VIII. Conclusions

In this paper, we have presented a new Security Framework for providing Cyber-security in Cloud Environments. A Capability_ID system has been developed as a mechanism by which the Framework could be implemented. The system is being developed to support e-Health applications that involve the monitoring of patients in their homes and storing the data in a Public Cloud environment.

References

[1] M. Clarke and C. Thiyagarajan, "A systematic review of technical evaluation in telemedicine systems," *Telemedicine and e-Health*, vol. 14(2), pp. 170–183, 2008.

[2] P. Honer, "Cloud Computing Security Requirements and Solutions: a Systemat6ic Literature Review," 2013.

[3] F. D, "Specifying reusable security requirements," *Journal of Object Technology*, vol. 3, no. 1, pp. 61 – 75, 2004.

[4] M. Aiash, G. Mapp, R. Phan, A. Lasebae, and J. Loo, "A Formally Verified Device Protocol using Casper/FDR," in *Proceedings of the 11th IEEE Conference on Trust, Security and Privacy in Computing and Communication (TrustCom 2012)*, June 2012.

[5] Mapp, G and Riley, L and Padily, A, "yRFC2: The Simple Protocol." [Online]. Available: http://www.mdx.ac.uk/research/science_technology/informatics/projects/ycomm.aspx(Lastaccessed:01/12/2013)

[6] M. V. Wilkes and R. M. Needham, "The Cambridge CAP Computer and its Operating System," in *Operating and Programming Series*. Elsevier North Holland, 1979.

[7] A. Tanenbaum and G. J. Sharp, "The Amoeba Distributed Operating System." [Online]. Available: www.cs.vu.nl/pub/amoeba/Intro.pdf

[8] R. Hinden and S. Deering, *RFC 4291 IP Version 6 Addressing Architecture*, IETF, February 2006.

[9] N. Doraswamy and D. Harkins, "IPSec: The New Security Standard for the Internet, Intranets and Virtual Private Networks," in *Prentice Hall Security Series*. Prentice Hall, 2003.

[10] G. Mapp, F. Shaikh, J. Crowcroft, D. Cottingham, and J. Baliosian, "Y-Comm: A Global Architecture for Heterogeneous Networking (Invited Paper)," in *3rd Annual International Wireless Internet Conference (WICON)*, October 2007.

[11] K. Masahiro, Y. Mariko, O. Ryoji, K. Shinsaku, and T. Tanaka, "Secure service and network framework for mobile ethernet," *Wireless Personal Communications*, vol. 29, 2004.

[12] R. J. Atkinson and S. N. Bhatti, *Identifier-Locator Network Protocol (ILNP) Architectural Description*, IETF, November 2012.

[13] P. Radkov, L. Yin, P. Goyal, P. Sarkar, and P. Shenoy, "A Performance Comparison of NFS and iSCSI for IP-Networked Storage," in *Proceedings of 3rd Usenix Conference on File and Storage Technologies (FAST 2004)*, April 2004.

[14] Solarlare, "10 Gbps NIC Cards for Sale." [Online]. Available: storage.dpie.com/products/solarflare/

[15] K. Kammueller, G. Mapp, S. Patel, and S. Abubaker, "Engineering Security Protocols with Model Checking - Radius-SHA256 and Secured Simple Protocol," in *Proceedings of the 7th International Conference on Internet Monitoring and Protection*, 2012.