33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

Secure Communications for PACS in a Cloud Environment

Tim Rostrom, Chia-Chi Teng

Abstract—**Picture Archiving and Communication Systems (PACS) have been traditionally constrained to the premises of the healthcare provider. This has limited the availability of these systems in many parts of the world and mandated major costs in infrastructure for those who employ them. Public cloud services could be a solution that eases the cost of ownership and provide greater flexibility for PACS implementations. Moving these systems to the public cloud requires that an authentication and encryption policy for communications is established within the PACS environment. This paper investigated an implementation which uses Transport Layer Security for communications between a cloud-based PACS server and client.**

I. Introduction

MEDICAL imaging systems have been traditionally constrained to the premises of the healthcare provider. These facilities incur major costs to provide the infrastructure for the medical imaging systems. Cost of ownership has been a major road block to small scale healthcare providers and in less developed areas. According to the World Health Organization, two-thirds of the world's population has no access to basic diagnostic imaging services [1]. These services are primarily unavailable because of insufficient infrastructure, unstable political environment and a considerable burden of disease. On the other hand, the increasing volume of diagnostic images in developed regions presents a different challenge. It is estimated that in 2014 healthcare providers in the US will perform over one billion diagnostic imaging procedures and generate approximately 100 Petabytes of data [2]. The amount of digital data being collected is leading to scalability and management issues for many healthcare providers.

Cloud computing provides an environment where services can be rapidly scaled up or down while costs incur only on a 'pay per use' basis without upfront capital costs. Real monetary saving can come from utilizing cloud computing for both small and large organizations [3], [4]. Other benefits include more robust cost-effective business continuity planning such as disaster recovery [2], and allowing more focus to be put on providing healthcare services than managing infrastructure [4].

These benefits do not come without risks. Maintaining the security and integrity of the data with a cloud environment becomes a major concern [5]. Legal policies concerning cloud computing are still being explored and debated [6]. Extending a medical imaging system from a protected network on the healthcare provider's premises to a public cloud service requires additional security measures including a secure communications policy that is carefully designed and implemented to protect data in transit. The security of data traveling over the open Internet is critical to protect patient privacy and the integrity of the data.

This paper discusses how secure communications can be established in a cloud-based medical image network with details outlined as the following. First, the current systems, standards and publications will be discussed. Second, the working prototype which creates secured communications with a cloud-based server. Lastly, an outline of future work that needs to be done will be presented.

II. Background Research

Picture Archiving and Communication Systems (PACS) are commonly used in the hospital environment as the tool to manage medical images. These systems have standardized on the Digital Imaging and Communications in Medicine (DICOM) file format and communications standard. Part 15 of the DICOM standard specifies Secure Transport Connection Profiles which includes two profiles which use certificates to establish a secure transport session, but details about how the secured connection is established and authenticated is left open to the application entity [7]. There are few publications which discuss how to establish a secure connection within a PACS environment let alone a PACS deployed to a public cloud. An authentication procedure in traditional healthcare systems is less of a concern given that most communications are on a private network behind a firewall and contained within the healthcare facility. Many implementations have a PACS router if communication outside of the protected network is needed. As we extend the system to utilize public cloud computing resources, establishing an authentication policy for secured communications is a necessity.

Cloud computing is still a developing industry where benefits and concerns are still being explored. Rosenthal et al [5] evaluates how cloud computing could be used for the healthcare industry. Some benefits discussed are reduced management decisions concerning infrastructure, scalability, increased resiliency and cost reductions. Even within a cloud environment, security management is still principally the responsibility of organization and is not outsourced to the cloud provider. Some additional considerations for organizations when moving to the cloud include the jurisdiction the cloud application will be under, additional risk of hackers and protecting data from the cloud provider and other tenants using cloud.

T. Rostrom is with Brigham Young University, Provo, UT 84602, USA. (e-mail: trostrom@byu.edu).

C. Teng is with Brigham Young University, Provo, UT 84602, USA. (phone: 801-422-1297; e-mail: ccteng@byu.edu).

Buyya et al [8] analyzes the trends of cloud computing and how they might be used by industry. Also included is an analysis of cloud computing infrastructure and some of the leading commercial cloud providers including Amazon EC2 [9], Google AppEngine [10], and Windows Azure [11].

The European Network and Information Security Agency (ENISA) [12] have published an extensive security analysis for the cloud and provided recommendations to manage and mitigate cloud specific risk. Many benefits to information security within a cloud environment were discussed including security on a large scale, rapid, smart scaling of resources and how service level agreements (SLA) force better risk management. Some risks inherent to cloud environments include vender lock-in, possible loss of governance and cloud service termination.

These evaluations, and many others, provide insights into the benefits and concerns regarding security within cloud services. With cloud computing being a relatively new industry, there is still much that needs to be discussed concerning how the services should be used and security measures put in place. Implementation of these security measures for medical imaging systems has not yet been widely discussed.

This research was to investigate the feasibility of secured DICOM communications with a cloud-based PACS implementation. To create a prototype for this project, we are leveraging an existing Windows Azure based DICOM server [13] and a mobile DICOM client for portable Ultrasound imaging [14]. Both of the projects implemented the standard DICOM networking protocol without the secured transport specification [7].

III. System Overview

Typical PACS has no mechanism for client authentication by username and password. Therefore, the client needs to validate the identity of the server and vice versa by using security certificates or other similar methods. DICOM specifies the requirement of secured data transmission in its Part 15 specification [7]. However, it does not specify a mechanism for authentication except stating that it is up to the application entity which should follow the transport layer security (TLS) or integrated secure communication layer (ISCL) standards. After a secured connection is established through either of these protocols, data will then be transferred according to the negotiated encryption method.

The standard used for this project is the Basic TLS Secure Transport Connection Profile specified in Part 15 of the DICOM standard [7]. With this profile, the TLS two-way client-server authentication via certificate exchanges [15] will be used. A unique certificate must be created and distributed to both the server and the client. When a certificate is created, it contains a private and public key for identification and encryption. These keys work in an asynchronous nature where data encrypted by one key can only be decrypted by the other. Only the owner of the certificate has possession of the private key. The public key is to be given freely to those who try to authenticate and securely communicate with the owner. Likewise, there are two versions of each certificate: private and public. The private certificate holds both the private and public keys and is only given to the owner. The public certificate only holds the public key and is given as part of the identification process of the owner. In this way, data encrypted by the public key can only be decrypted by the private key and vice versa. This is used for both encryption and as proof of identity during the authentication procedure.

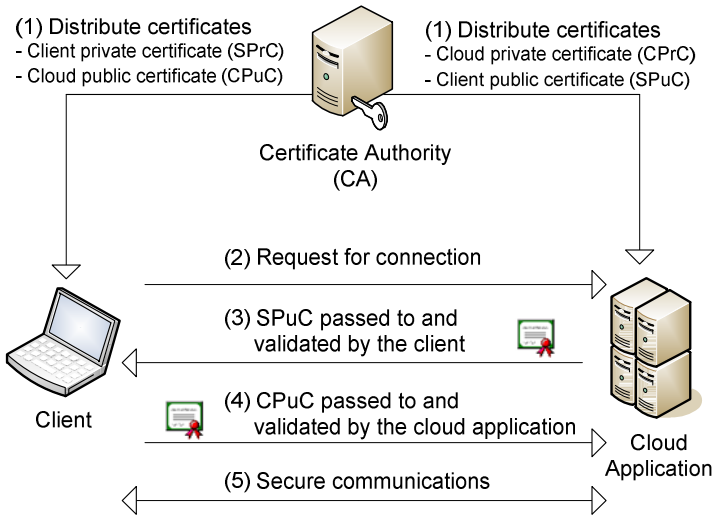

Fig. 1. Transport Layer Security authentication procedure

A. *Certificate Creation and Distribution*

A trusted certificate authority (CA) will create a certificate for both the cloud application and the client. This CA may be a public authentication service like Verisign (www.verisign.com) or an enterprise CA that is controlled by the cloud-based PACS service. The cloud server will hold both the server private certificate (SPrC) and the client public certificate (CPuC). Likewise, the client will hold the client private certificate (CPrC) and the cloud server public certificate (SPuC). These certificates will be used to provide encryption and identification during the authentication process.

For this prototype system, we used the Windows Internet Information Services Manager to generate the certificates. These are self-signed certificates meaning that the CA is also the owner of the certificate. Certificates issued by public authentication services or an enterprise CA will also work in this prototype.

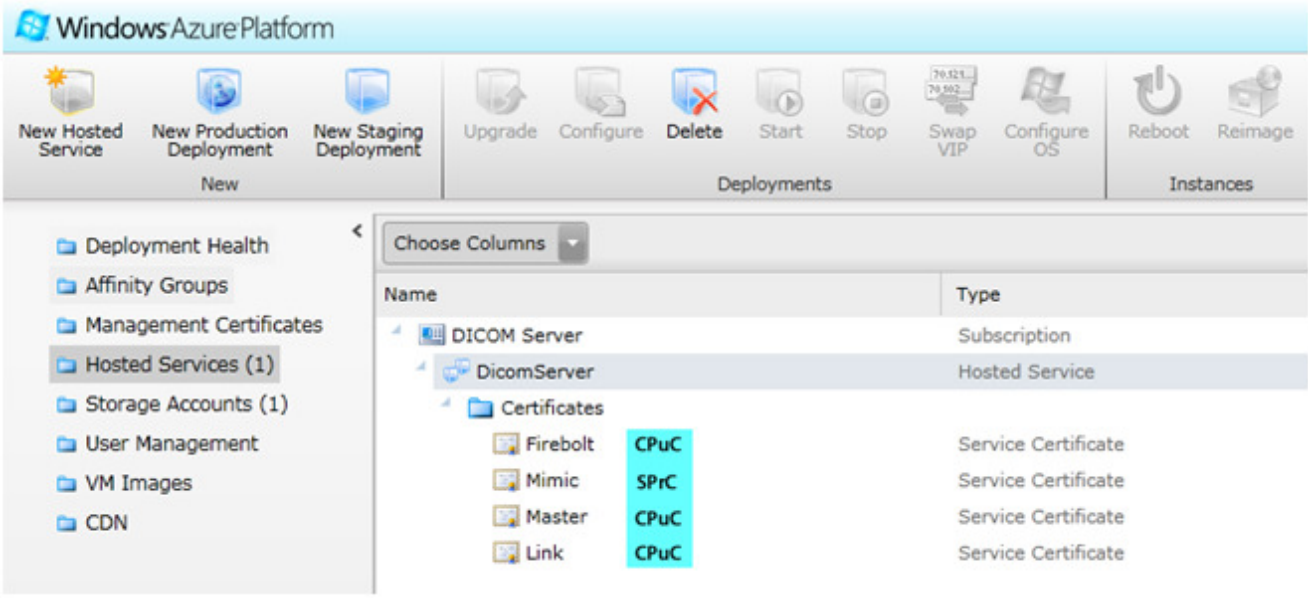

Fig. 2. Certificates imported into the Windows Azure certificate manager

B. *Authentication*

Figure 1 shows the high-level TLS two-way authentication procedure. The simplified process is as follows,

1) Creation & distribution of certificates by the trusted CA to setup both cloud application and client.
2) The client requests for a connection from the server.
3) The server responds with its public certificate, SPuC. The client checks if the SPuC identifies the trusted cloud server.
4) If the SPuC passes the test, the client responds with its public certificate, CPuC. The server verifies that the CPuC is from a trusted client according to the validation procedure.
5) If the CPuC passes the test, a symmetric key is used to encrypt the remainder of the communication session.

C. *Cloud Implementation*

While the transport layer security protocol is standardized, operating systems (OS) have varying degree of built-in support regarding tools to manage the security certificates and application programming interface (API) to access them. OpenSSL library [16] was created to provide a cross-platform unified tools and API to manage and access certificates, but it is still difficult to use and more importantly lacks sophisticated management tools with graphical user interface (GUI). Microsoft Windows OS has a built-in certificate management system with a GUI application which makes it easier to manage certificates. The .NET library also provides an easy to use API to access certificates to establish secured transport connections.

Microsoft's Windows Azure has built-in .NET support similar to the desktop OS, which makes it easy to access the certificates through the high level API. Certificates are added to the cloud service through the Azure web interface, as shown in Figure 2. This online Azure certificate manager allows x.509 certificates [17] to be uploaded to be used by the cloud applications. Using Visual Studio with the Azure SDK, certificates available in the Azure certificate manager can be linked to specific applications. These certificates can be used for proof of identity or validation of client certificates.

For the cloud DICOM server, a single certificate, SPrC, is used for a cloud deployment to provide proof of the server's identity to the client. Also, all the client public certificates, CPuC, are installed to the certificate manager. When a client requests a connection to the cloud server, the server responds by passing its public certificate, SPuC, to the client. The client then uses the previously installed, SPuC, to validate the certificate it received from the cloud server.

After the client has validated the identity of the cloud server, the client presents its public certificate, CPuC, for proof of identity to the cloud server. The server checks the CPuC against the list of trusted client certificates received from the CA. This procedure assures that only authorized clients can connect to the cloud server.

D. *Client Implementation*

The DICOM client used was created for the Microsoft Windows OS written with the .NET framework. Figure 3 shows the Windows Certificate Manager which controls all personal and trusted CA certificates. The CPrC and SPuC certificates are installed to this manager. When the client receives the SPuC from the cloud application, the client validates the certificate against the trusted SPuC received from the CA. After a successful validation, the client sends its public certificate, CPuC, to the server for identification.

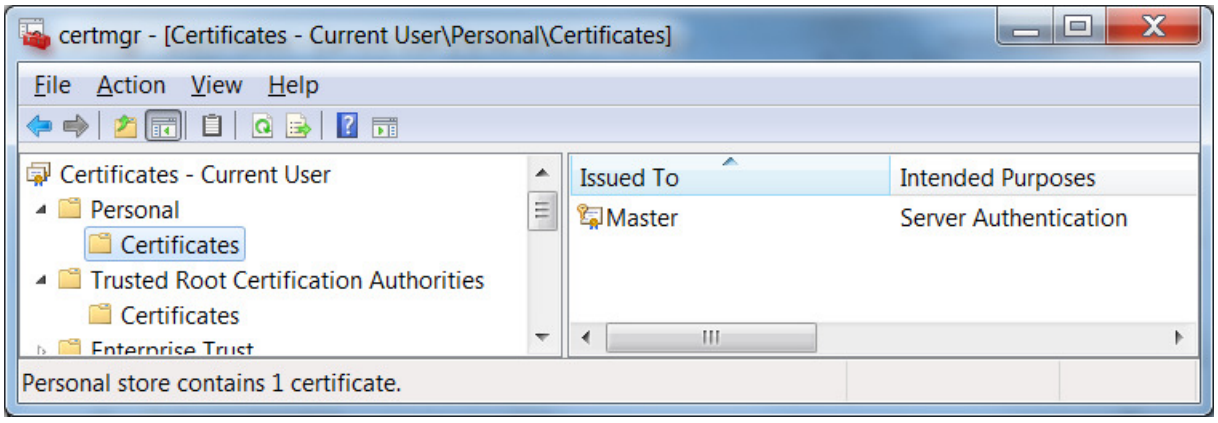

Fig. 3. Windows Certificate Manager containing the client's private certificate

IV. Results

This prototype implemented a secure connection between a cloud-based DICOM server and client. The Windows Azure and .NET platforms provided all the necessary tools for the authentication and encryption functionality. As discussed in previous section, Azure's certificate management service works with .NET to execute these processes. Server certificates can be added or removed while the cloud application is running, and they can be shared among multiple instances of the applications.

The secured DICOM transport was examined with the Wireshark network packet analyzer (http://www.wireshark.org). The authentication process can prevent typical spoofing attack. The communication session was confirmed to be encrypted and safe from man-in-the-middle attack. We also tested the difference in transmission speed between using secure communications and not. The tests were performed by uploading a DICOM image to the cloud server using the C-STORE message and measuring the time it took to complete the protocol. We found that it took an average of 7 seconds when the communication was not secured and 9 seconds when it was secured. This shows the expected increase in time resulted from the overhead of securing the communication.

V. Future Work and Conclusions

The implementation of this prototype demonstrates that a medical imaging server placed on public cloud services can authenticate and secure the communications with its clients as required by HIPAA rules. This prototype is a proof of concept and therefore needs some work to become a more practical implementation. Work needs to be done to create a certificate management policy which will allow for a more scalable and flexible solution. This policy needs to include certificate creation, distribution, authentication and account for groups or organizations of clients.

Moving medical imaging servers to the cloud enables healthcare providers to extend their reach with mobile clients that can function anywhere the internet can be accessed. It also provides the benefits of cloud computing including scalability, pay per use and reduction of infrastructure management. This can benefit large and small healthcare providers throughout the world to provide better diagnostics imaging services, reduce costs and focus more on providing healthcare services than infrastructure management.

References

[1] WHO report, "Essential diagnostic imaging," World Health Organization, http://www.who.int/eht/en/DiagnosticImagin.pdf
[2] "Prepare for disaster & tackle terabytes when evaluating medical image archiving," Frost & Sullivan, http://www.frost.com, 2008.
[3] M. Armbrust, .A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski et al, "Above the clouds: a Berkeley view of cloud computing," EECS Department, University of California, Berkeley Technical Report, Feb. 2009.
[4] A. Rosenthal, P. Mork, M. H. Li, J. Stanford, D. Koester, P. Reynolds, "Cloud computing: a new business paradigm for biomedical information sharing," *Journal of Biomedical Informatics,* vol. 43, pp. 342-353.
[5] J. Harauz, L. M. Kaufman, B. Potter, "Data security in the world of cloud computing," *IEEE Security & Privacy*, July 2009.
[6] P. T. Jaeger, T. Lin and J. M. Grimes, "Cloud computing and information policy: computing in a policy cloud". Journal of Information Technology & Politics, vol 5(3), pp. 269-283, 2008.
[7] *Digital Imaging and Communications in Medicine (DICOM)*, National Electronical Manufacturers Association, Rosslyn, VA.
[8] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems,* vol. 25, pp. 599-616.
[9] Amazon elastic compute cloud (EC2). http://www.amazon.com/ec2/
[10] Google app engine. http://appengine.google.com
[11] Microsoft window azure. http://www.microsoft.com/windowsazure/
[12] "Cloud computing: benefits, risks and recommendations for information security," ENISA, Nov. 2009.
[13] C. C. Teng, J. Mitchell, C. Walker, A. Swan, C. Davila, D. Howard, T. Needham, "A medical image archive solution in the cloud," Software Engineering and Service Sciences IEEE International Conference, 2010.
[14] C. C. Teng, C. Green, R. Johnson, P. Jones, C. Treasure, "Mobile ultrasound with DICOM and cloud connectivity," IEEE 2010 Congress on Services (SERVICES 2011), in press.
[15] T. Dierks, E. Rescorla, "RFC 5246 – the transport layer security (TLS) protocol version 1.2," IETF, Network Working Group, Aug. 2008.
[16] OpenSSL cryptography and SSL/TLS toolkit. http://www.openssl.org.
[17] R. Housley, W. Ford, W. Polk, D. Solo, "RFC for x.509 – internet x.509 public key infrastructure: certificate and crl profile," IETF, Network Working Group, Jan. 1999.