High Assurance Challenges for Cloud Based Computing

Coimbatore Chandersekaran, William R Simpson, and Ryan R Wagner

***Abstract*—Cloud computing is emerging as an attractive, cost effective computing paradigm. However, many of the applications require high assurance, attribution and formal access control processes including defense, banking, credit, content distribution, etc. Current implementations of cloud services do not meet high assurance requirements. The high assurance requirement presents many challenges to normal computing and some rather precise requirements that have developed from high assurance issues for web service applications. The challenges of high assurance associated with cloud computing are primarily in four areas. The first is virtualization and the loss of attribution that accompanies a highly virtualized environment. The second is the loss of ability to perform end-to-end communications. The third is the extent to which encryption is needed and the need for a comprehensive key management process for public key infrastructure, as well as session and other cryptologic keys. The fourth is in monitoring and logging for attribution, compliance and data forensics. We explore each of these challenges and discuss how they may be able to be overcome. Our view of high assurance and the issues associated with web services is shaped by our work with DoD and the Air Force, but applies to a broader range of applications, including content delivery and rights management.**

***Index Terms*—Attribution, Cloud Computing, IT Security, Virtualization.**

I. INTRODUCTION

CLOUD computing must has come to mean many different things. To some, it is simply putting one's data on a remote server. However, in this paper, we utilize the definition provided by NIST [24]. They define five essential characteristics of any cloud computing environment:

1. On demand self-service,
2. Broad network access,
3. Resource pooling,
4. Rapid elasticity, and
5. Measured service.

It is important to note that multi-tenancy and virtualization are *not* essential characteristics for cloud computing. For our discussion we will assume no multi-tenancy and virtualization, since the latter adds the most efficiency. Arguments below do not require either. Cloud computing is, at its core, a *service*. There are three primary models of this service. In the lowest level Infrastructure as a Service (IaaS), storage, computation, and networking are provided by the cloud provider to the cloud consumer. In the next level up of Platform as a Service (PaaS), all of the trappings of IaaS plus an operating system and perhaps some application programming interfaces (APIs) are provided and managed by the cloud provider. The highest service model is Software as a Service (SaaS), in which the cloud provider provides an end-user service such as webmail. The higher the service model, the more control the cloud provider has as compared to the cloud consumer. There are four different models for deploying cloud services. Primarily, they are public or private clouds. In a public cloud, the infrastructure--although generally not the data on it--may be used by anyone willing to agree to its terms of use. Public clouds exist off the premises of the cloud consumer. Private cloud infrastructure is used only by one organization. It may exist either on or off the organization's premises. There are two twists to these infrastructures. In a community cloud, a group of organizations with similar interests or needs share a cloud infrastructure. That infrastructure is not open to the general public. In a hybrid cloud, two or more cloud deployment models are connected in a way that allows data or services to move between them. An example of this would be an organization's private cloud that makes use of a community cloud during loads of high utilization.

II. BENEFITS OF THE CLOUD

Cloud computing benefits emerge from economies of scale [25]. Large cloud environments with multiple users are better able to balance heavy loads, since it is unlikely that a large proportion of cloud consumers will have simultaneously high utilization needs. The cloud environment can therefore run at a higher overall utilization, resulting in better cost effectiveness. In a large cloud computing environment, rather than having a number of information technology generalists, the staff has the ability to specialize and become the masters of their own domains. In many cloud environments this balancing is done by virtualization and the use of a hypervisor. With regard to information security, the staff can become even more specialized and spend more time hardening platforms to secure them from attacks. In the homogeneous cloud environment, patches can be rolled out quickly to the nearly identical hosts.

A. Drawbacks of the Cloud

Cloud computing is not without its drawbacks. In cases where services are outsourced, there is a degree of loss of

Manuscript received June 6, 2011; revised July 23, 2011. This work was supported in part by the U.S. Secretary of the Air Force and The Institute for Defense Analyses. The publication of this paper does not indicate endorsement by the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations

Coimbatore Chandersekaran is with the Institute for Defense Analyses.(email: cchander@ida.org)
William R. Simpson is with the Institute for Defense Analyses, 4850 Mark Center Drive, Alexandria, Virginia 22311 USA and is the corresponding author phone: 703-845-6637, FAX: 703-845-6848 (e-mail: rsimpson@ida.org)
Ryan Wagner is with the Institute for Defense Analyses.(email: rwagner@ida.org)

control. This can affect compliance with laws, regulations, and organizational policies. Cloud systems have additional levels of complexity to handle intra-cloud communications, scalability, data abstraction, and more. To be available to cloud consumers, cloud providers may need to make their services available via the Internet. And critically, many clouds use multi-tenancy, in which multiple organizations simultaneously utilize a single host and virtualization. If one tenant organization is compromised or malicious, it may be able to compromise the data or applications of the other organizations on the same host. The load balancing may use a single identity for all instances of a service whether it is virtual or real.

B. *Some Changes in the Threat Scenario*

There are clear differences in many of the threat scenarios as detailed below [26]:

1. Loss of governance (or visibility and/or control of the governance process)
2. Lock-in (threats may be present and locked into the cloud environment).
3. Isolation failure (e.g., hypervisor attack, lack of accountability)
4. Compliance risks (if provider cannot provide compliance evidence or will not permit audit by customer, lack of accountability)
5. Management interface compromise (and or inheritance of threats and/or malicious code from other users of the cloud).
6. Data protection (how does customer verify protection, lack of accountability)
7. Insecure or incomplete data deletion
8. Malicious insider (often the cloud insider is not vetted as well as the organizational insider, and insiders from other customers could bring in contagious viruses – see 5 above.)

C. *Differences from Traditional Data Centers*

Cloud computing relies on much of the same technical infrastructure (e.g., routers, switches, operating systems, databases, web servers) as traditional data centers and as a result, many of the security issues are similar in the two environments. The notable exception in some cases is the addition of a hypervisor for managing virtual machines. The Cloud Security Alliance's security guidance states "Security controls in cloud computing are, for the most part, no different than security controls in any IT environment. Cloud computing is about gracefully losing control while maintaining accountability even if the operational responsibility falls upon one or more third parties." While many of the controls are similar, there are two factors at work that make cloud computing different: perimeter removal and trust. With cloud computing, the concept of a network or information perimeter changes radically. Data and applications flow from cloud to cloud via gateways along the cloud perimeters. However, since the data may be stored in clouds outside the organization's premises or control, perimeter controls become less useful. In exchange for the lack of a single perimeter around one's data and applications, cloud consumers must be able to trust their cloud providers. A lack of trust in a cloud provider does not necessarily imply a lack of security in the provider's service. A cloud provider may be acceptably secure, but the novelty of cloud computing means that many providers have not had the opportunity to satisfactorily demonstrate their security in a way that earns the trust of cloud consumers. Trust must be managed through detailed Service Level Agreements (SLAs), with clear metrics and monitoring mechanisms, and clear delineation of security mechanisms [27].

III. HIGH ASSURANCE COMPUTING

While the current implementations of Cloud Computing provide efficient and operationally friendly solutions to data computing and content distribution, they are not up to the challenge of high assurance.

In certain enterprises, the network is continually under attack. Examples might be:

- Banking industry enterprise such as a clearing house for electronic transactions,
- Defense industry applications,
- Credit card consolidation processes that handle sensitive data both fiscal and personal,
- Medical with concerns for privacy and statutory requirements,
- Content Distributor's worried about rights in data, or theft of content.

The attacks have been pervasive and continue to the point that nefarious code may be present, even when regular monitoring and system sweeps clean up readily apparent malware. This Omni-present threat leads to a healthy paranoia of resistance to observation, intercept and masquerading. Despite this attack environment, the web interface is the best way to provide access to many of its users. One way to continue operating in this environment is to not only know and vet your users, but also your software and devices. Even that has limitations when dealing with the voluminous threat environment. Today we regularly construct seamless encrypted communications between machines through SSL or other TLS. These do not cover the "last mile" between the machine and the user (or service) on one end, and the machine and the service on the other end. This last mile is particularly important when we assume that malware may exist on either machine, opening the transactions to exploits for eaves dropping, ex-filtration, session high-jacking, data corruption, man-in-the-middle, masquerade, blocking or termination of service, and other nefarious behavior. Before we examine the challenges of Cloud Computing systems, let us first examine what high assurance architecture might look like.

A. *Basic Tenets*

This section provides nine tenets that guide decisions in an architectural formulation for high assurance and implementation approaches [12]. These tenets are separate from the "functional requirements" of a specific component (e.g., a name needs to be unique); they relate more to the goals of the solution that guide its implementation.

- The *zeroth* tenet is that the *Malicious entities* can look at all network traffic and send virus software to network assets. In other words, rogue agents (including insider threats) may be present and to the extent possible, we should be able to operate in their presence, although this does not exclude their ability to view some activity. Assets are constantly monitored and cleaned, however new attacks may be successful at any time and nefarious code may be present at any given time.

- The *first* tenet is *simplicity*. This seems obvious, but it is notable how often this principle is ignored in the quest to design solutions with more and more features. That being said, there is a level of complexity that must be handled for security purposes and implementations should not overly simplify the problem for simplicity's sake.
- The *second* tenet, and closely related to the first is *extensibility*. Any construct we put in place for an enclave should be extensible to the domain and the enterprise, and ultimately to cross-enterprise and coalition. It is undesirable to work a point solution or custom approach for any of these levels.
- The *third* tenet is *information hiding*. Essentially, information hiding involves only revealing the minimum set of information to the outside world needed for making effective, authorized use of a capability. It also involves implementation and process hiding so that this information cannot be farmed for information or used for mischief.
- The *fourth* tenet is *accountability*. In this context, accountability means being able to unambiguously identify and track what active entity in the enterprise performed any particular operation (e.g. accessed a file or IP address, invoked a service). Active entities include people, machines, and software process, all of which are named registered and credentialed. By accountability we mean attribution with supporting evidence. Without a delegation model, and detailed logging it is impossible to establish a chain of custody or do effective forensic analysis to investigate security incidents.
- Th**is** *fifth* tenet is *minimal detail* (to only add detail to the solution to the required level). This combines the principles of simplicity and information hiding, and preserves flexibility of implementation at lower levels. For example, adding too much detail to the access solution while all of the other IA components are still being elaborated may result in wasted work when the solution has to be adapted or retrofitted later.
- The *sixth* is the emphasis on a *service-driven* rather than a product-driven solution whenever possible. Using services makes possible the flexibility, modularity, and composition of more powerful capabilities. Product-driven solutions tend to be more closely tied to specific vendors and proprietary products. That said, commercial off-the-shelf (COTS) products that are as open as possible will be emphasized and should produce cost efficiencies. This means that for acquisition functionality and compatibility are specified as opposed to must operate in a Microsoft forest [18] environment.
- The *seventh* tenet is that *lines of authority* should be preserved and IA decisions should be made by policy and/or agreement at the appropriate level.
- The *eighth* tenet is *need-to-share* as overriding the need-to-know. Often effective health, defense, and finance rely upon and are ineffective without shared information.

B. Architectural Features

In order to build an architecture that conforms to these tenets, there must be elements that insure that they are built into the systems. In the architecture we espouse, the basic formulation follows a web 2.0 approach and uses Organization for the Advancement of Structured Information Standards (OASIS) standards of security [4]. These elements are listed below:

Naming and Identity

Identity will be established by the requesting agency. In the DoD this is primarily through the Electronic Data Interchange Personal Identifier (EDIPI), but for other certificate authorities, their naming scheme must be honored. To avoid collision with the EDIPI, the identity used by all federated exchanges shall be the distinguished name as it appears on the primary credential provided by the certificate authority. The distinguished name must be unique over time and space which means that retired names are not reused and ambiguities are eliminated. Naming must be applied to all active entities (persons, machines, and software).

Credentials

Credentials are an integral part of the federation schema. Each identity (all active entities) requiring access shall be credentialed by a trusted credentialing authority. Further, a Security Token Server (STS) must be used for storing attributes associated with access control. The STS that will be used for generating Security Assertion Markup language (SAML) tokens must also be credentialed (primarily through the same credentialing authority, although others may be entertained.

PKI required – X.509 Certificates

The primary exchange medium for setting up authentication of identities and setting up cryptographic flows is the Public Key Infrastructure (PKI) embodied in an X.509 certificate.

Certificate Services

The certificate authority must use known and registered (or in specific cases defined) certificate revocation and currency checking software.

Bi-Lateral End-to-End Authentication

The requestor will not only authenticate to the service (not the server), but the service will authenticate to the requestor. This two way authentication avoids a number of threat vulnerabilities. The requestor will initially authenticate to the server and set up a Secure Socket Layer (SSL) connection to begin communication with the service. The primary method of authentication will be through the use of public keys in the X.509 certificate, which can then be used to set up encrypted communications, (either by X.509 keys or a generated session key). The preferred method of communication is secure messaging, contained in Simple Object Access Profile (SOAP) envelopes. All messages are encrypted for delivering to the recipient of the message.

Authorization Using SAML Packages

All authorizations will be through the use of SAML packages in accordance with the SAML 2.0 specification provided by OASIS [5].

Registration of the STS

All STS that create and sign SAML packages must be registered. The certificate of the STS will be used to sign SAML tokens, and complete bi-lateral authentication between requestors and the STS.

Recognizing STS Signatures

STS signatures will be recognized only for registered STSs and may be repackaged by the local STS when such registration has been accomplished. Unrecognized signatures will not be honored and the refusal will be logged as a security relevant event.

Certificate Caches

Local STSs within the enterprise forests will maintain a certificate cache of all registered STSs to facilitate the re-issuance of SAML packages when appropriate.

IV. CHALLENGES IN BRINGING THE CLOUD AND HIGH ASSURANCE TOGETHER

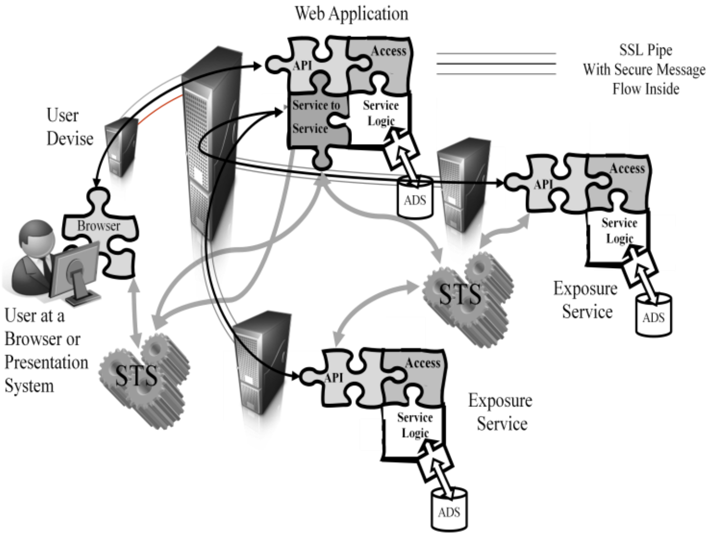

Fig. 1. High Assurance Security Flows

Despite the obvious advantages of cloud computing, the large amount of virtualization and redirection poses a number of problems for high assurance. In order to understand this, let's examine a security flow in a high assurance system.

The application system consists of a web application (for communication with the user), one or more aggregation services that invoke one or more exposure services and combines their information for return to the web application and the user, As a pre-requisite to end-to-end communication an SSL or other suitable TLS is setup between each of the machines.

The exposure services retrieve information from one or more Authoritative Data Sources (ADSs). Each communication link in Fig. 1 will be authenticated end- to-end with the use of public keys in the X.509 certificates provided for each of the active entities.

This two way authentication avoids a number of threat vulnerabilities. The requestor initially authenticates to the service provider. Once the authentication is completed, an SSL connection is established between the requestor and the service provider, within which a WS-Security package will be sent to the service. The WS-Security [7, 10] package contains a SAML token generated by the Security Token Server (STS) in the requestor domain. The primary method of authentication will be through the use of public keys in the X.509 certificate, which can then be used to set up encrypted communications, (either by X.509 keys or a generated session key). Session keys and certificate keys need to be robust and sufficiently protected to prevent malware exploitation. The preferred method of communication is secure messaging using WS Security, contained in SOAP envelopes. The encryption key used is the public key of the target (or a mutually derived session key), ensuring only the target can interpret the communication.

The problem of scale-up and performance is the issue that makes cloud environments and virtualization so attractive. The cloud will bring on assets as needed and retire them as needed. Let us first examine scale-up in the unclouded secure environment. We will show only the web application, although the same rules apply to all of the communication links between any active elements shown in the fig. above. The simplest form of dividing the load is to stand up multiple independent instances and divide users into groups who will use the various instances. Dependent instances that extend the thread capabilities of the server are considered single independent instances. Remember, all independent instances are uniquely named and credentialed and provisioned in the attribute stores. A representation that is closer to the cloud environment is shown in Fig. 2.

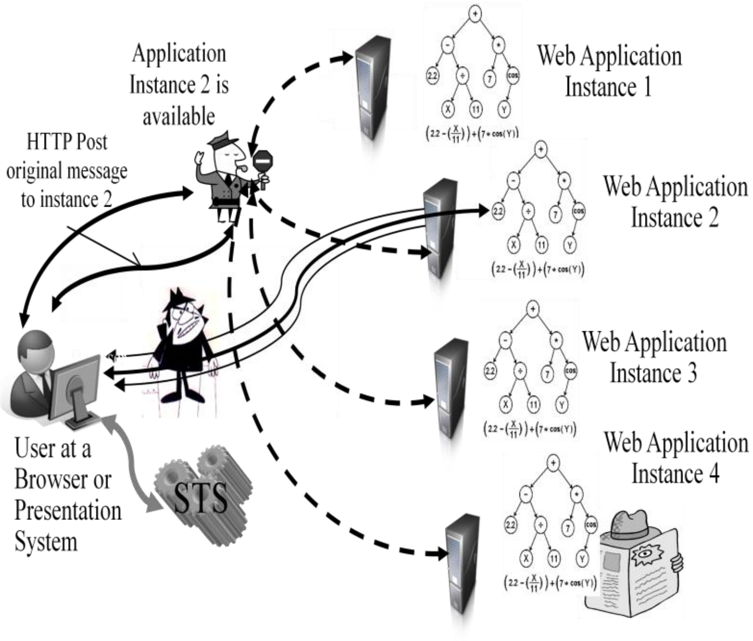
Application Instance 2 is available
HTTP Post original message to instance 2
Web Application Instance 1
Web Application Instance 2
Web Application Instance 3
Web Application Instance 4
User at a Browser or Presentation System
STS
(22-(X/11))+(7*cos(Y))

Fig. 2. High Assurance Load Balancing

A traffic cop (load balancer) monitors activity and posts a connection to an available instance. In this case all works out since the new instance has a unique name, end-point, and credentials with which to proceed. All of this, of course needs to be logged in a standard form and parameters passed to make it easy to reconstruct for forensics. We have shown a couple of threats that need mitigation where one eavesdrops on the communication and may actually try to insert himself into the conversation (man-in-the-middle). This highlights the importance of bi-lateral authentication and encrypted communications. The second is present on instance 4 and highlights the need to protect caches and memory spaces.

When a cloud environment runs out of resources for computing, it builds additional instances, some of these may be thread extension schemas, and some may be independent instances. The traffic cop here is often called a hypervisor and it keeps track of the instances and connections. Fig. 3 shows notionally how this operation works. When thread capacity is saturated at the server, the hypervisor would nominally redirect the request to an independent virtual or real instance of the web application. If none exists, it will build one from elements in the resource pool as depicted in instance 4 on the chart. If the last user signs off of an independent virtual or real instance (instance 3 in the fig.), the hypervisor tears down the instance and places the resources back into the resource pool. This provides an efficient re-allocation of resources.

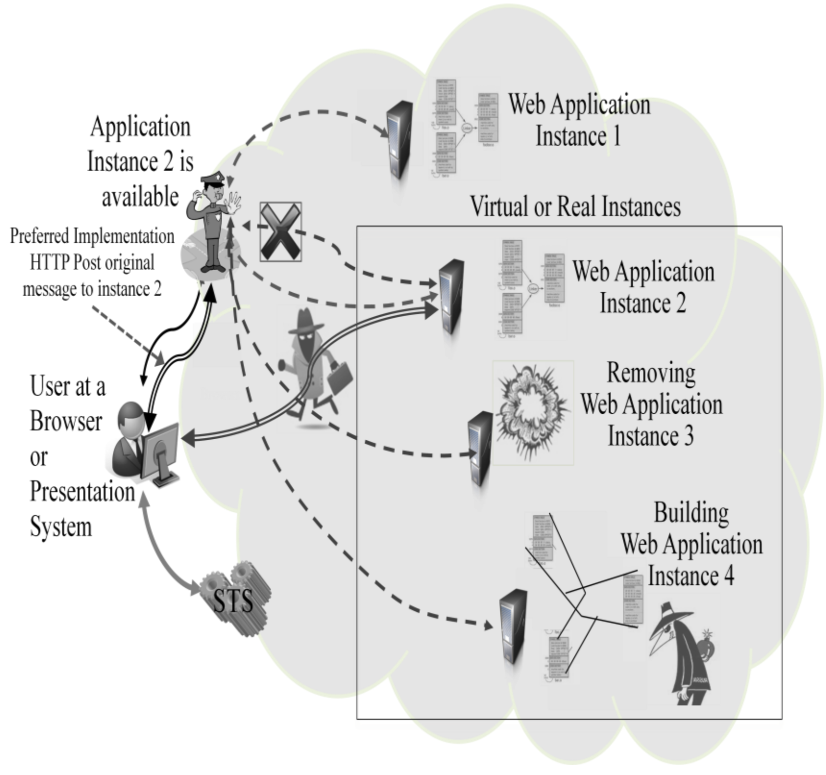
Application Instance 2 is available
Preferred Implementation HTTP Post original message to instance 2
User at a Browser or Presentation System
STS
Web Application Instance 1
Virtual or Real Instances
Web Application Instance 2
Removing Web Application Instance 3
Building Web Application Instance 4

Fig. 3. High Assurance Virtualized Hypervisor Activity

There are several steps that must be taken to preserve the security, if we are interested in a high assurance computing environment. The number of independent instances must be anticipated. Names, credentials and end points must be assigned for their use. The attribute stores and HSMs must be provisioned with properties and key to be used. The simple re-direct must be changed to a re-post loop as in Fig. 2. The user will then have a credentialed application to authenticate with bi-laterally and an end point for end-to-end message encryption. Key management is complex and essential. When a new independent instance is required it must be built, and activated (credentials and properties in the attribute store, as well as end point assignment). All of these activities must be logged in a standard format with reference values that make it easy to reassemble the chain of events for forensics. When a current independent instance is retired, it must be disassembled, and de-activated (credentials and properties in the attribute store, as well as end point assignment).

All of these activities must be logged in a standard format with reference values that make it easy to reassemble the chain of events for forensics. The same threats exist, and the same safeguards must be taken. In fact, in Fig. 3 nefarious code is built right into the virtual or real instance 4, which underscores the need for trusted and verified software to do the virtualization, and protection of the resources while they are in the resource pool.

A recap of these challenges is listed below:

1. Shared Identities and credentials break the accountability paradigm.
   - Each independent instance of a virtual or real machine or virtual or real service must be uniquely named [20] and provided a PKI Certificate for authentication. The Certificate must be activated while the virtual machine is in being, and de-activated when it is not, preventing hijacking of the certificate by nefarious activities. The naming and certificates must be pre-issued and self-certification is not allowed. Each instance of an independent virtual or real machine or virtual or real service must have a unique end point. This may take some manipulation through the load balancing process but is required by attribution and accountability. This means that simple re-direct will not work. Extensions of the thread mechanism by assigning resources to the operating system may preserve this functionality. The individual mechanism for virtualization will determine whether this can be accomplished.
2. Multi-tenancy (multiple tenants using a single host) must be prohibited. This does not mean that multiple instances of an application cannot use the same host. The latter may be an acceptable extension of the thread count.
3. No virtualization across machines (each virtual machine must reside in a single real machine). Protecting resources across more than one machine is problematical.
4. Each potential independent instance of a service must have an account provisioned with appropriate elements in an attribute store. These must be pre-issued and linked to the unique name for each potential instance of a service. This is required for SAML token issuance.
5. A cloud based Security Token Service (STS) needs to be installed and implemented and it must meet all of the requirements listed her for uniqueness of names and end points as well as instantiated certificates and

cryptographic capability. The STS is considered trusted software and may be load balanced in the traditional sense, using a single set of credentials and provisioned attributes.

6. The importance of cryptography cannot be overstated, and all internal communications as well as external communications should be encrypted to the end point of the communication. Memory and storage should also be encrypted to prevent theft of cached data and security parameters.
7. Private keys must reside in Hardware Storage Modules (HSMs). The security of the java software key store does not meet high assurance criteria.
    - Stand-up of an independent virtual or real machine or virtual or real service must link keys in HSM, and activate credentials pre-assigned to the virtual service.
    - Stand-down of an independent virtual or real machine or virtual or real service must de-link keys in HSM, and de-activate credentials pre-assigned to the virtual service.
    - Key Management in the virtual environment is a particular concern and a complete management schema including destruction of session keys must be developed.
8. Proxies and re-directs break the end-to-end paradigm. When end points must change, a re-posting of communication is the preferred method. There must be true end-to-end communication with full attribution. This will mean that communication must be re-initiated from client to server when a new virtual or real instance is instantiated, it must have a unique end point, with unique credentials and cryptography capabilities.
9. Resource pools must be protected from persistent malicious code.
10. All activities must be logged in a standard format with reference values that make it easy to reassemble the chain of events for forensics.

The aforementioned challenges are daunting, but provisions must be made if high assurance computing environments are take advantage of the cloud computing environment.

V. SUMMARY

We have reviewed the basic approaches to clouds and their potentials for savings in computing environments. We have also discussed at least one high assurance architecture and its' requirements which provide direct challenges to the way cloud computing environments are organized. Notably the extensive use of virtualization and re-direction is severe enough that many customers who need high assurance have moved away from the concept of cloud computing [21 - 23]. We believe, however, that a precise statement of the high assurance requirements will lend themselves to solutions in the cloud computing environment, and expand the potentials use of this technology.

REFERENCES

[1] Burrows, M. and Abadi, M and Needham, R, M., "A Logic of Authentication," *ACM Transaction on Computer Systems*, vol. 8, no. 1, pp. 18-36, 1990.

[2] Needham R.M., and Schroeder, R. M., 'Using Encryption for Authentication in Large Networks of Computers', *Communications of the ACM*, vol. 21, no. 12, pp. 993-999, 1978.

[3] Internet. Shibboleth Project, Available at http://shibboleth.internet2.edu/. (Accessed on 19 February 2011)

[4] OASIS Identity Federation, *Liberty Alliance Project*, Available at http://projectliberty.org/resources/specifications.php. (Accessed on 19 February 2011)

[5] OASIS Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, Available at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security. (Accessed on 19 February 2011)

[6] NIST Special Publication 800-95, *"Guide to Secure Web Services,"* Available at: http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf. (Accessed on 19 February 2011)

[7] "Web Service Security: Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0", Microsoft Corporation, 20k six pages05

[8] "WS-ReliableMessaging Specification", OASIS, June 2007

[9] "WS-SecureConversation Specification", OASIS, March 2007

[10] "WSE 3.0 and WS-ReliableMessaging", Microsoft White Paper, June 2005, Available at http://msdn2.microsoft.com/en-us/library/ms996942(d=printer).aspx. (Accessed on 19 February 2011)

[11] FIPS PUB 196, Federal Information Processing Standards Publication, "*Entity Authentication Using Public Key Cryptography*", February 18, 1997

[12] Air Force Information Assurance Strategy Team, *Air Force Information Assurance Enterprise Architecture*, Version 1.70, SAF/XC, 15 March 2009. [Not available to all]

[13] Foster, I. and Kesselman, C. and Tsudik, G and S. Tuecke, A Security Architecture for Computational Grids, *Proceedings of the 5th ACM Conference on Computer and Communications Security Conference*, pp. 83-92, 1998.

[14] Welch V, and Foster,. I and Kesselman, C and Mulmo, O and Pearlman and L Tuecke, S. and Gawor, J and Meder, S and Siebenlist F., X.509 Proxy Certificates for Dynamic Delegation., *3rd Annual PKI R&D Workshop*, 2004.

[15] Belani, E. and Vahdat, A. and Anderson, T. and Dahlin, M., The CRISIS wide area security architecture, In *Usenix Security Symposium*, January 1998.

[18] *Windows Server 2003: Active Directory Infrastructure*. Microsoft Press. 2003. pp. 1–8 to 1–9. ISBN: 0-7356-1438-5

[19] Lewis, M and Grimshaw, A., "The Core Legion Object Model," In *Proceedings of the 5th IEEE Symposium. On High Performance Distributed Computing*, Pages 562-571. IEEE Computer Society Press, 1996.

[20] Standard for Naming Active Entities on DoD IT Networks, Version 3.5, September 23, 2010

[21] Remarks-Debra Chrapaty, Corporate Vice President, Global Foundation Services, Microsoft Mgt Summit, Las Vegas, May 2008, http://www.microsoft.com/Presspass/exec/debrac/mms2008.mspx (accessed 19 February 2011).

[22] Bobbie Johnson, technology correspondent, guardian.co.uk, Cloud computing is a trap, warns GNU founder Richard Stallman, 29 September 2008, http://www.guardian.co.uk/technology/2008/sep/29/cloud.computing.richard.stallman (accessed 19 February 2011).

[23] Andy Plesser, Executive Producer, Beet.tv, Cloud Computing is Hyped and Overblown, Forrester's Frank Gillett.....Big Tech Companies Have "Cloud Envy", http://www.beet.tv/2008/09/cloud-computing.html, September 26, 2008 (Accessed on 19 February 2011)

[24] Peter Mell, Timothy Grance, NIST SP 800-145 Draft: Cloud Computing, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, January 2011, http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf

[25] Wayne Jansen, Timothy Grance, NIST SP 800-144 Draft: Guidelines on Security and Privacy in Public Cloud Computing, Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, January 2011, http://csrc.nist.gov/publications/drafts/800-144/Draft-SP-800-144_cloud-computing.pdf

[26] Daniele Catteddu and Giles Hogben, European Network Information Security Agency (ENISA), Cloud Computing Risk Assessment, November 2009, http://www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment

[27] Cloud Security Alliance, Security Guidance for Critical Areas of Focus in Cloud Computing V2.1, December 2009, https://cloudsecurityalliance.org/csaguide.pdf