IMPLEMENTING X.509 SECURITY CERTIFICATE BASED AUTHENTICATION IN A VIRTUAL ORGANIZATION

A Thesis

Presented to

The Faculty of the Graduate School

At the University of Missouri

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

VEERENDRA SHIROLE

Dr. Gordon K. Springer, Thesis Supervisor

JULY 2011

The undersigned, appointed by the dean of the Graduate School,

have examined the thesis entitled

IMPLEMENTING X.509 SECURITY CERTIFICATE BASED AUTHENTICATION

IN A VIRTUAL ORGANIZATION

Presented by Veerendra Shirole

A candidate for the degree of

Master of Science

And hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Gordon Springer

Dr. Youssef Saab

Dr. Jianlin Cheng

ACKNOWLEDGEMENTS

I would like to sincerely thank my advisor, Dr. Gordon Springer, for his unwavering support throughout my entire research. Without his guidance and constructive feedback, the completion of this thesis would have not been possible. He has continuously motivated me and inspired my growth as a student of Computer Science at the University of Missouri. I would also like to thank Dr. Youssef Saab and Dr. Jianlin Cheng for taking time out of their schedule to be on my thesis committee and to review my research material. I would like to thank all the research assistants and staff members of the Research Support Computing Group at the University of Missouri for their valuable inputs and suggestions.

Finally, I would like to express my heartfelt gratitude to everyone who has directly or indirectly contributed to my academic and personal development at the University of Missouri.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ABSTRACT

In recent years, research institutions have shown great interest in sharing computing resources with other institutions as a cost effective alternative to maintaining dedicated resources for computationally intensive tasks. A set of such research institutions forming an inter-institutional research environment is referred to as a Virtual Organization (VO). However, a Virtual Organization is not limited to research institutions. Any two or more parties with an interest in collaborative research can create a Virtual Organization. An important function of such a Virtual Organization is to authenticate users at all participating computing sites, without the users having to maintain an array of authentication credentials. Shibboleth is an infrastructure used to provide the facility of single sign-on in many Virtual Organizations. However, every participating institution in a Virtual Organization may not have a Shibboleth authentication mechanism for its users. A Shibboleth infrastructure implementation is a huge challenge that entails conformance to the policies of the institution that participates in the Virtual Organization. Therefore in the absence of a Shibboleth authentication mechanism at a user's home institution, it may not be possible for users of a participating institution to access resources belonging to other institutions in the Virtual Organization.

This thesis addresses the issue of authenticating users who do not necessarily possess Shibboleth credentials, but are authentic users that need access to the resources in a Virtual Organization. In order to authenticate such users, there is a need to involve an alternative method of authentication that does not rely on a username/password credential provided by any particular institution that is a part of the Shibboleth enabled infrastructure. Lately, X.509 security certificates have gained immense popularity as a method for verifying the identity of a person. These certificates can be used to authenticate users on any system that trusts the certificate's signing Certificate Authority. Incorporating support for certificate-based authentication in the Shibboleth infrastructure enables the Virtual Organization to authenticate users that belong to the research environment, but do not necessarily have Shibboleth credentials. Certificate-based authentication can also provide resource access to guests of participating institutions that include, but are not limited to, visiting faculty or consultants to the participating institutions. Thus, certificate-based authentication increases the resource providing capability of the research environment by servicing all the users that are entitled to use resources in the Virtual Organization.

CHAPTER 1 - INTRODUCTION

The last few years have seen a substantial increase in the need for resource sharing in the higher education research community. As people from various organizations and institutions collaborate to work on projects of shared interests, a need arises to share resources belonging to each other to jointly work on projects in a cost-effective way [1]. A Virtual Organization is such a federation consisting of two or more entities that have come together to work on projects of common interest. This collaboration helps researchers by providing resources from more than one institution to fulfill their research requirements. However, every institution has a security system that requires a user to provide a credential for authentication prior to accessing some institutional resources. Hence in a resource-sharing collaborative environment, the user needs to acquire a username and password from the resource owning institution in order to authenticate for accessing protected resources or services. A major task faced by a Virtual Organization is implementing an authentication and authorization process that helps the user access resources belonging to various institutions without maintaining an array of credentials.

The Great Plains Network (GPN) [2] consortium defines a Virtual Organization, for example, that uses Shibboleth [3] with some extensions, as an infrastructure for authenticating and authorizing users requesting access to resources. The

Shibboleth system is an open source software system for single sign-on in a Virtual Organization. In the Shibboleth architecture, a user is authenticated by an Identity Provider (IdP) at the home institution of that user [4]. An Identity Provider attests the identity of a user after verifying the user's credentials. In the case that a user does not belong to a participating institution, the user needs to acquire valid credentials from an institution with a Shibboleth IDP to access the resources. Examples of users that do not belong to any participating institution are visiting faculty or consultants to institutions in the Virtual Organization. Additionally, in a Virtual Organization, the users and resources may cross institutional boundaries making the problem of authorizing users more complex for decision making. An Entitlement Server [1] is implemented as an extension to Shibboleth that allows a Virtual Organization, such as GPN, to make informed authorization decisions for users requesting access to resources [3]. Resource providing institutions define entitlements on resources that users need to possess to access resources. However, these entitlements are not the same as user attributes. While user attributes like name, address and user affiliation are associated with a person, entitlements are given to resources provided by the Virtual Organization. Organizational resources are not a part of user attributes but somehow must be assigned to users so that the users can access resources. For this, an identity management system is not sufficient to empower users to access resources. Hence a separate Entitlement Server records what users are entitled to access the defined resource. These resources may in fact belong to a Service Provider that is completely separate from institutions with identity management systems. Authorization decisions are made based on these entitlements assigned to the user and are subject to the access policy of the institution that provides the resources. An entitlement is an attribute whose value decides whether a user is entitled to access a resource or service. Only after successful authentication and proper authorization is the user able to access the requested resource. This process aims at preventing unauthorized access of the computing resources in the Virtual Organization.

Using Shibboleth, with an Entitlement Server, as a middleware architecture simplifies identity management and permissions for providing access to resources. A Shibboleth Identity Provider provides information about user identity and the Entitlement Server determines user authorization for accessing any resource. Implementation of Shibboleth at a home institution requires conformance to a set of policies and rules of the concerned home institution. This conformance may lead to a delay in implementation of Shibboleth at many institutions that are a part of a Virtual Organization. Users belonging to such home institutions that do not have a Shibboleth Identity Provider cannot be authenticated within a Virtual Organization. This prevents users belonging to institutions that are a part of the Virtual Organization from accessing computing resources, unless the users acquire credentials from another institution with an accepted Shibboleth Identity Provider. However, in case of Virtual Organizations where a large number of institutions do not have a Shibboleth Identity Provider, an alternative mechanism for authentication is needed. This thesis presents a different approach to authenticate users requesting access to resources in a Virtual Organization that can be used in addition to the existing institutional authentication mechanisms.

This thesis focuses on using X.509 Security Certificates [5] to ascertain the authenticity of a user requesting access to a computing resource. X.509 is a standard of a public key infrastructure (PKI) [6] for single sign-on and privilege management. A X.509 security certificate is a digital document assuring the identity of an individual/organization. This certificate is signed by a Certificate Authority (CA) [7] that is trusted by the verifying institution. Thus, a recognized certificate can be used to authenticate a user in a Virtual Organization. The proposed service of X.509 certificates-based authentication works in tandem with the conventional username and password authentication provided by a Shibboleth Identity Provider. The proposed implementation provides an option of authenticating a user with a valid security certificate. The identity of the authenticated user is then used to make authorization decisions to provide or deny access to resources requested by the user.

This thesis is organized into six chapters. In the next chapter, a detailed description of the Shibboleth architecture is provided. This chapter speaks about the authentication and authorization process in more detail. In Chapter 3, an explanation of the X.509 Security certificate-based authentication is provided. Public key certificate, certificate revocation lists and X.509 certificates are also discussed in this chapter. Chapter 4 provides a description of the design and architecture of the system implemented in this thesis. Chapter 5 provides details of the implementation of security certificate-based authentication and the integration of security certificate-based authentication with the Shibboleth authentication mechanism. Chapter 6 concludes the thesis by discussing pros and cons of the current implementation, results and possible applications of certificate-based authentication in other systems.

CHAPTER 2 – AUTHENTICATION AND AUTHORIZATION IN A VIRTUAL ORGANIZATION

This chapter describes the authentication and authorization infrastructure in a Virtual Organization. An Entitlement Server, which is a new component, when added to the Shibboleth system implements fine-grained authorization. This component is also discussed. Authentication of a user is a procedure for ensuring that the user is indeed who he/she claims to be. On the other hand, authorization is the process of determining whether the authenticated user has the authority to access a requested resource [1]. Finally an overview of the Shibboleth system and the Entitlement Server is presented to understand the need, design and implementation used in developing the code for this thesis.

2.1 What is Shibboleth?

A Virtual organization (VO) [8] is a collaboration among institutions that come together to work on projects of common interest or for sharing computational resources. The users in a Virtual Organization face various challenges during collaborative research for accessing computing resources or services. The users normally need to maintain multiple passwords for accessing resources belonging to different institutions. The Virtual Organizations, on the other hand, have to control access to resources provided by the service providers to provide a secure environment. To address the challenges faced during authentication of a user, a

Virtual Organization can use Shibboleth as a mechanism for authentication. Shibboleth is an open-source [9] single sign-on[1] [10] system developed as a part of the Internet2 Middleware Initiative [11] that focuses on tackling various challenges faced by Virtual Organizations.

Shibboleth provides a single sign-on and attribute exchange framework using the Security Assertion Markup Language[2] (SAML) [12]. Users authenticating in the Virtual Organization using Shibboleth need only one log-on credential as opposed to one per resource [13]. The policy driven authentication mechanism of Shibboleth enables institutions to maintain control over their data while protecting user privacy. Shibboleth also provides the capability allowing the user and the identity providing institution to control the attributes released to applications. The resource hosting organizations do not need to store and maintain user data known to the user's organization, thus protecting the user privacy. The user's organization is in total control of the user information released to the resource provider [13]. Management of identity and permissions for collaborative research environments is thus simplified by the use of Shibboleth enabled access.

[1]Single sign-on (SSO) is a mechanism wherein a single action of user authentication permits a user to access all computers and systems where the user identity can be used for authentication, without the need to enter multiple username/password combinations.
[2] SAML is a standard used for exchanging authentication and authorization information among various entities in a federated environment.

2.2 *Basic Shibboleth Architecture*

The web-based single sign-on mechanism of Shibboleth is the result of an Identity Provider (IdP) and a Service Provider (SP) that work together to achieve a single sign-on system. A detailed description of the working and key features of the basic Shibboleth system can be found in [4]. An Identity Provider authenticates users and provides user information to a Service Provider. The IdP is located at an organization maintaining a user's account. This organization is also referred to as the user's home institution. The Service Provider works with an IdP to provide access to resources according to rules and policies of the resource owning institution. The SP is usually located at the resource hosting organization. The following diagram shows the rudimentary Shibboleth authentication and authorization mechanism.

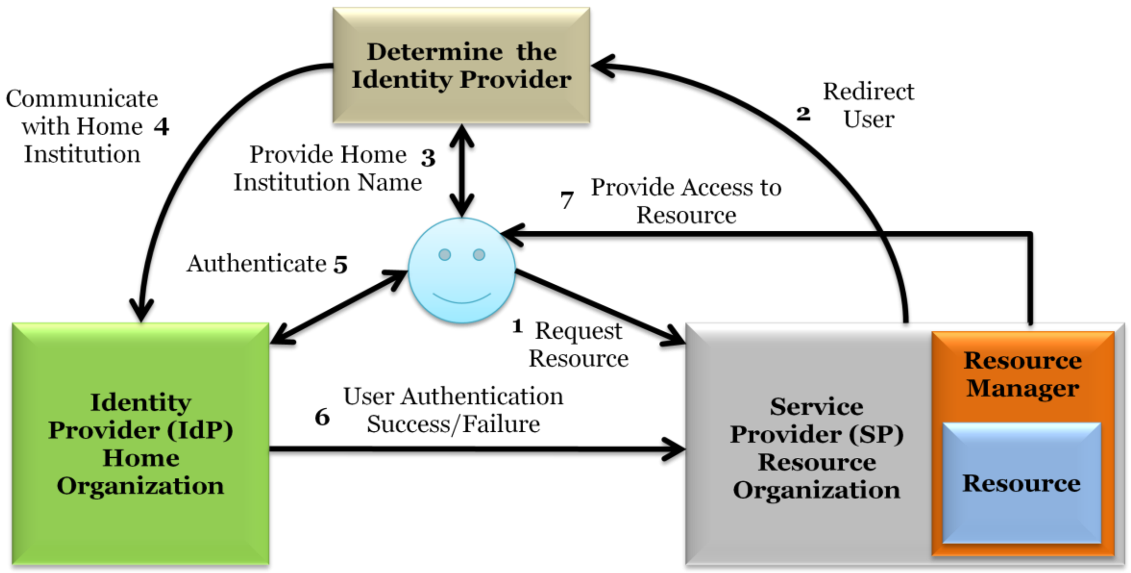

Determine the Identity Provider
Communicate with Home 4 Institution
Redirect User 2
Provide Home 3 Institution Name
7 Provide Access to Resource
Authenticate 5
1 Request Resource
Identity Provider (IdP) Home Organization
6 User Authentication Success/Failure
Service Provider (SP) Resource Organization
Resource Manager
Resource

Figure 1: Basic Shibboleth Architecture[3]

[3] Reference: Understanding Shibboleth: Basic Interaction https://spaces.internet2.edu/display/SHIB2/NewUnderstandingShibboleth

A user requests access to a resource. The SP receives the requests. If the user is not authenticated, the user needs to be authenticated. The Identity Provider of the user is determined by querying the user's Home Institution. The user and the authentication request are sent to the user's IdP. The user provides authentication credentials to the IdP. The user is authenticated by the IdP and sent back to the SP along with the authentication response specifying whether the user is authentic or not. The authentication response is verified by the SP and if successful, the user's request is sent to the resource that provides service to the user.

This basic Shibboleth architecture is common for most Shibboleth implementations. A description of above architecture can be also found in [14]. However, every Virtual Organization has a set of requirements and goals that require this implementation to be customized as per the needs of the Virtual Organization. The following section provides an in-depth description of the modified Shibboleth implementation for fine-grained authorization in a Virtual Organization.

2.3 Authentication and Authorization Using Shibboleth and an Entitlement Server

The Shibboleth environment with the Entitlement Server extension is presented in this section. Shibboleth is responsible for authenticating users in a Virtual

Organization. However, for fine-grained authorization of a user, Shibboleth can be integrated with an Entitlement Server to provide a fine-grained authorization mechanism. To better understand the working of authentication and authorization using Shibboleth and an Entitlement Server, an understanding of the Entitlement Server and various components of the Shibboleth enabled environment and is necessary.

The Shibboleth Identity Provider (IdP) is a component of the Shibboleth infrastructure used for user authentication. The IdP is a part of the Identity Management System of the user's home institution. The Shibboleth infrastructure lets users of an organization use the organization's existing authentication mechanism to authenticate and request access to resources. When a user requests access to a resource, the Service Provider (SP) redirects the user to authenticate at the home institution. The user provides authentication credentials to establish the user's identity. The IdP authenticates the user using the home institution's authentication mechanism. Once a user is authenticated, the identity of the user is established and the SP can make access control decisions about the user to determine if the user is to be granted access to resources.

The Shibboleth Service Provider (SP) is a component that provides access to a resource. A user sends a resource access request to the service provider that hosts the resource. The SP has to ensure that this user has the appropriate privilege to access the resource. An unauthenticated user is directed to the IdP for authentication. After the user is successfully authenticated, the SP checks if the user is authorized to access the resource by querying the Entitlement Server (ES) with the entitlement required to access the requested resource. Based on the whether the entitlement query to the ES fails or succeeds, the SP makes an informed access control decision to grant access to the requested resource or not.

The Entitlement Server (ES) is an additional component implemented as an extension to the Shibboleth architecture to make fine-grained authorization decisions in a Virtual Organization. The Entitlement Server answers entitlement queries made by the SP to authorize user requests. However, the Service Provider needs to authorize itself with the Entitlement Server before making any entitlement requests. An entitlement is an attribute that is used to allow or restrict user access to a specific resource or group of resources. The Service Provider works in conjunction with the Entitlement Server to make fine-grained access control decisions for users requesting access to resources hosted by the Service Provider. The Entitlement Server is hosted as a secure server accessible to any authorized Service Provider within a Virtual Organization.

When a user requests access to a resource hosted by a SP, the SP needs to redirect the user to the user's home institution. However, for this redirection, the SP needs to be aware of the user's home institution. Since there are many institutions that could be part of the Virtual Organization, an additional component is needed to identify the home institutions that belong to the Virtual

Organization. This component is known as the Identity Provider Discovery. The current Shibboleth implementation used in this project uses a module known as the WAYF (Where Are You From) service to redirect users from the SP to the IdP. The WAYF service is a static web-page with a drop-down list of all the Shibboleth Identity Providers acceptable to the Service Provider. On this web-page, the user selects his/her respective home institution and is redirected to the institution's authentication page. Thus the WAYF service is responsible for directing the users to their respective IdP's for authentication.

2.3.1 The Authentication and Authorization Protocol

Figure 2 presents the step by step processing of the authentication and authorization protocol. A user requests access to a resource provided by the SP. Before providing access to the resource, the SP has to make sure that the user is authentic and has authorization to access the requested resource. Thus, the first task of the SP is to ensure that the user is authenticated. In case of a user that is not authenticated, the SP redirects the user to the Identity Provider Discovery (WAYF service in case of this project). The user specifies the home institution that the user belongs to. The Identity Provider Discovery communicates with the home institution and the user is provided the authentication page of his/her home institution. The user then authenticates at the home institution. Currently, the only method available for user authentication is username/password based authentication via an institution's IdP. This thesis provides an alternative by supporting authentication using X.509 Security Certificates [5]. In case of a successful authentication, the IdP provides a unique handle for the user to the SP. The SP uses this handle to request additional attributes from the IdP. These requests are serviced by the IdP so that the SP can make authorization decisions. However, in addition to making attribute inquires to the IdP, the SP also sends entitlement queries to the ES. The ES provides a YES/NO answer back to the SP. By combining the information retrieved from the IdP and the ES, the SP verifies the attributes and the entitlements with the needed access policy rules to make fine-grained authorization decisions for user requests to access the resources. A more in-depth description of the Shibboleth architecture and its integration with the Entitlement Server can be found in [1].

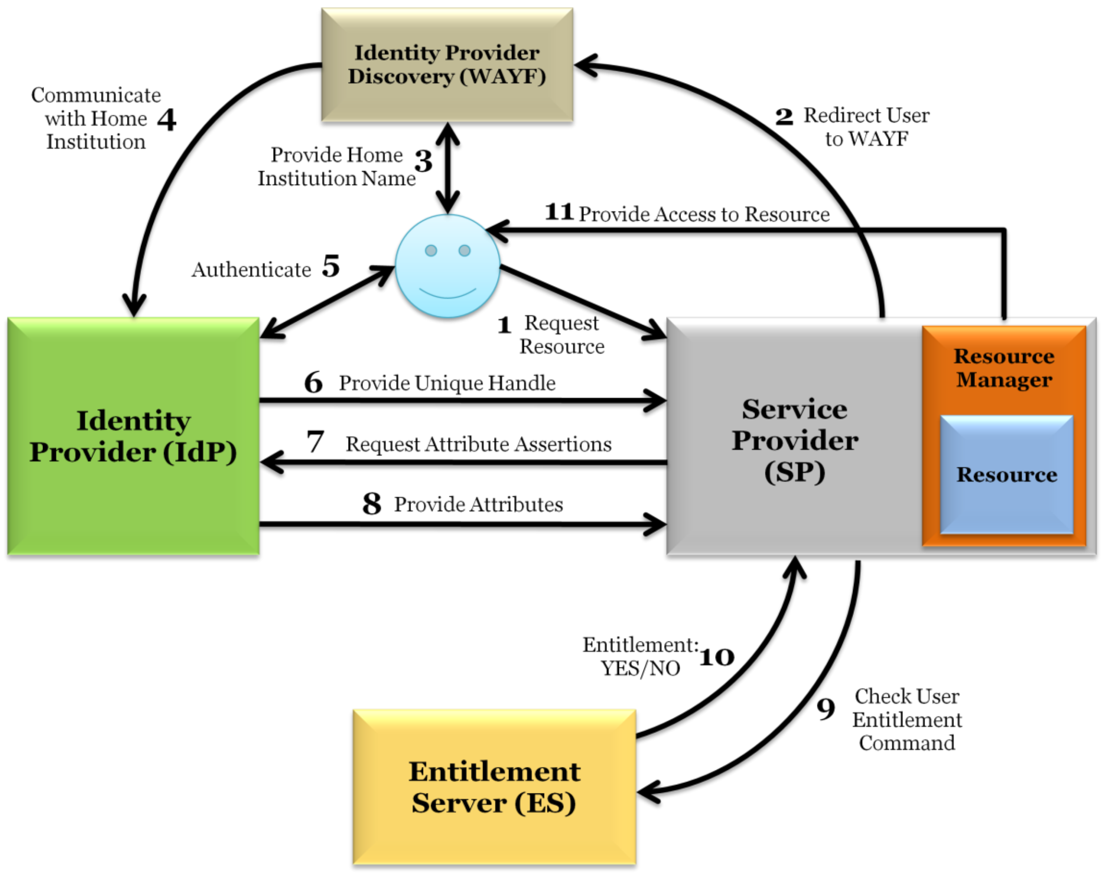


Figure 2: The Authentication and Authorization Protocol

This chapter provides a basic description of the Shibboleth architecture and the Entitlement Server that is used for fine grained authorization. The next chapter focuses on X.509 Security Certificates which is a different approach to authentication as compared to the username/password authentication supported by Shibboleth.

CHAPTER 3 - X.509 SECURITY CERTIFICATES

This chapter explains the need for an alternative to username/password based authentication in the Shibboleth environment. The feasibility of using security certificates as an alternate method for Shibboleth authentication in the Virtual Organization environment is discussed. The basic concept of Public Key Certificates is described and followed by a detailed description of the X.509 security certificates.

3.1 Need for Security Certificate Based Authentication

While the Shibboleth infrastructure and the Entitlement Server successfully authenticate and authorize in a Virtual Organization, this infrastructure suffers from some shortcomings. The most impacting of these shortcomings is that home institutions without a Shibboleth-enabled Identity Provider have no way of authenticating their users in the Shibboleth environment. There are also cases when the Virtual Organization is required to provide username/password credentials to users, like visiting faculty or consultants,that are temporarily associated to an institution. The maintenance of these credentials is complicated by the temporary nature of some of these users' affiliations. To accommodate the requirements of valid users without access to a Shibboleth IdP, alternatives to the traditional username/password authentication need to be considered.

A notable alternative to the username/password authentication is authentication using a digital security certificate. Digital certificates are widely used to authenticate users, institutions or even web-sites. Digital certificates can be verified against well known certificate authorities by any verifying entity. In fact, Shibboleth uses digital certificates for communication between Service Providers and Identity Providers [15]. The Service Provider uses Public Key Infrastructure (PKI) to authenticate with an Identity Provider and to establish a secure connection. Similarly, the PKI infrastructure also can be used for authentication of users in a Virtual Organizations that possess a security certificate signed by an acceptable certificate authority.

3.2 Public Key Certificates

According to Request For Comments (RFC) 2828 [16], a certificate refers to "*a document that attests to the truth of something or the ownership of something*." In the context of a digital certificate, a certificate refers to a digitally signed record containing a name and a public key [17]. A certificate is used to assign a public key to a person, website, device or any other entity and is used to demonstrate the lawful ownership of a public key [7]. These certificates are also called public key certificates. As per RFC 2828 a public key certificate is a special kind of a certificate "*that binds a system entity's identity to a public key value, and possibly to additional data items.*" As per [7] a public key certificate is a digitally signed data structure that demonstrates the true ownership of a public key. In order to obtain a security certificate, an individual or organization requests a certificate from a Certificate Authority (CA). Certificate Authorities are authorities that are recognized and trusted by a community of users. The CA attests the certificate by signing it and producing a public key certificate for the requester. This public key certificate can thus be used for authenticating the identity of the certificate owner by any individual or organization that recognizes and trusts the CA attesting the security certificate.

Figure 3 shows the structure of a Public Key Certificate. A Public Key Certificate is comprised of at least a public key, some naming information and one or more digital signatures [7]. The public key section is the public key of the certificate owner. The certificate exists to establish the authenticity of this public key. The naming information identifies the owner of the certificate and the public key. The digital signature section is a MD5[4] [18] hash of the previous certificate data signed using the CA's private key. Such an infrastructure in which a CA binds a public key to an entity is known as a Public Key Infrastructure (PKI). PKI is usually referred to as a foundation on which other components of applications, systems and network security are built [6]. However, PKI is not limited to certificate management. PKI also includes archive management, key management and token management for a community of entities that employ public key cryptography and can therefore be used to issue, revoke and validate public keys and public key certificates [7].

[4] MD5 is an algorithm that takes an input message of arbitrary length and produces a 128-bit "message digest" or "fingerprint" of the input message.

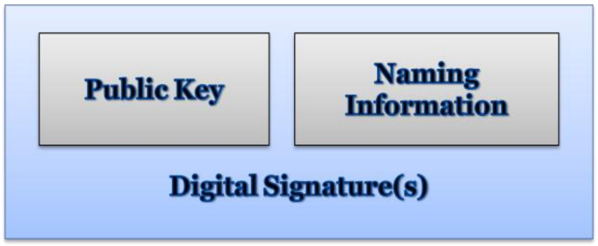


Figure 3: Structure of a Public Key Certificate [7]

3.3 *X.509 Certificates*

Today, one of the most relevant types of public certificates is the X.509 certificate [7]. The X.509 certificates conform to the International Telecommunication Union - Telecommunication Standardization Sector's [19] (ITU-T) X.509 recommendation. This recommendation specifies both a format for the certificate as well as a certificate distribution scheme [5]. The ITU-T X.509 recommendation was first published in 1988. The standard is currently at its third revision X.509 version 3 (X.509v3) that was officially released in 1996. The format of a X.509 certificate file with an example is provided in Appendix A.

The X.509 recommendation follows a hierarchical trust model for trusting the authenticity of a certificate. This means that a user must define a list of trusted CAs and certificates from which the trust can be extended. The CAs are logically divided into two groups, namely trusted root CAs and intermediate CAs. A trusted root CA is preconfigured to be trusted by default. The certificate belonging to a trusted root CA is self-signed which means that the subject and the issuer of the certificate are the same. Intermediate CAs on the other hand are not trusted by default, but are trusted if the certificates of the intermediate CAs are issued by trusted root CAs. This trust hierarchy can be extended to form a chain of intermediate CAs with a trusted root CA at the top level. This model is known as the hierarchical trust model that is used for handling certification chains [7].

Another important advantage of using X.509 certificates for authentication is the ability to revoke certificates ahead of the expiration of the validity period of the certificate. In such cases when the certificates need to be revoked, the CA periodically issues a Certificate Revocation List (CRL) that contains a list of all the certificates that have been revoked. The web clients can thus reject certificates that have been listed in the CRLs. Due to the impracticality of handling enormously large CRLs some web browsers these days are able to retrieve validity information of a certificate online instead of looking up large revocation lists [7]. A detailed description of certificate revocation can be found in [20].

3.4 Authentication Using X.509 Certificates

The Shibboleth architecture implemented for this thesis uses the distinguished name of a user and the institution the user belongs to, to establish a user's unique identity. This information is enough to uniquely identify a user and determine whether the user has the necessary authorization to access a particular resource.

Thus the first step in authenticating a user using a certificate is to establish the unique identity of the user. The email address of the user contained within the certificate serves the purpose of establishing identity as the email address provides a distinguished name (username) and the institution of the user. This identity of the user can be used to make entitlement queries for authorization decisions. Thus it is essential that the certificate contains an email address that can be extracted to obtain a user's identity.

However, the possession of a security certificate does not guarantee that the user is the owner of the certificate. Especially since a X.509 certificate is a public certificate, any user with access to a certificate can pretend to be the owner of the certificate and request access to resources of the Virtual Organization. The private key of a user can be involved in the authentication procedure to validate the user, for example from the user's web browser where the private key is installed or by prompting the user to provide the private key. However, it is impractical for a user to be in possession of the private key under certain scenarios like when the user is authenticating from a public computer. Thus, to ensure that the user providing a X.509 certificate is indeed the owner of the certificate it is necessary to provide an extra degree of security. To achieve this protection, in the implementation of this thesis, a user authenticating using a X.509 certificate is required to answer a challenge question in lieu of providing the private key. A challenge question/answer is similar to a username/password combination that is chosen by the owner of the certificate and can be used anywhere and anytime, not just from the web browser. Correctly answering the challenge question confirms that the user is indeed the owner of the certificate and can be provided access to the resources. The procedure of authenticating users via certificates using a challenge question ensures that only authentic users are granted access to the resources. This challenge question requirement is as safe and secure as having a username/password query response scheme.

Authorizing a user to use certificate-based authentication entails creation of a challenge question for that particular user. This is achieved by maintaining a special entitlement for the users. This entitlement has the value "x5" and it denotes that the user has been authorized to use certificate-based authentication. This entitlement holds the challenge question for the user and the SHA-2 (Secure Hashing Algorithm; see Appendix C) hash of the answer to that question. A user can modify his/her challenge question only after successfully authenticating into the system. However, first-time addition of the "x5" entitlement can only be done by an administrator in the Virtual Organization or a "Certificate Administrator". A Certificate Administrator is an administrator that is appointed only for certificate-based authentication and has no extra privileges in the Virtual Organization. The administrator and certificate administrator can authorize certificate-based access by adding the "x5" entitlement for the user with a random question and answer. The user is required to modify the challenge question/answer during his/her first successful authentication into the system. The user can select one of a few pre-defined challenge questions or can create a new challenge question. The administrator and certificate administrator can also reset the challenge question/answer for the user in case the user forgets his/her password. Resetting a challenge question generates a random challenge answer that provides user access to the system and needs to be modified after first successful authentication. The administrators and certificate administrators authorize the users to use certificate-based authentication (by adding the "x5" entitlement) only after validating the email of the user. Users of the system can only use the certificates that contain the same email that the user holds the "x5" entitlement with. This ensures that the users can only authenticate using their personal certificates as opposed to a certificate provided to a server or a web site.

An advantage of authenticating users using a X.509 certificate is that the validity of a certificate can be controlled and thus a user can successfully authenticate using a certificate only for a specified time period. The validity of a certificate can be used as an extra measure of security for authenticating users. In case of authenticating guests of a Virtual Organization that need access to resources, the guests can be provided certificates that are valid only for the time period that the guests need access. This relieves administrators from the manual task of tracking invalid users and invalidating user credentials when the user is no longer associated with a Virtual Organization.

This chapter discussed the possibility of using security certificates as an alternative for authenticating users in the Shibboleth environment. Public Key

Certificates followed by the description of X.509 certificates that are used as a method of authentication in this thesis were discussed. With the information regarding the X.509 certificates and the architecture of Shibboleth discussed in the previous chapter, the next chapter proceeds to discuss the design of the architecture for authentication and authorization implemented in this thesis.

CHAPTER 4 – SYSTEM DESIGN AND ARCHITECTURE

This chapter presents the design and architecture of the system presented in this thesis. Initially the design considerations for the implementation of X.509 certificate based authentication are explained. The architecture of certificate-based authentication is presented followed by its integration in the existing mechanism to handle user authentication requests.

4.1 System Design Considerations

The certificate-based authentication mechanism presented in this thesis is designed to be modular so that it can be integrated easily with the existing username and password authentication mechanism. This design helps to implement certificate-based authentication with minimal changes to the existing system. Hence while the system may alternate between different modes of user authentication, certificate-based authentication and username/password based authentication, once the user has been authenticated the entitlement checking procedure remains unchanged.

This design focuses on reusing the existing infrastructure of Shibboleth with an Entitlement Server to minimize the new components implemented in the system.

As is explained in detail later in this chapter, this design uses the Entitlement Server/Client infrastructure as a server-client framework for certificate-based authentication in order to reduce the need for a separate server and client. The coupling of the Entitlement Server and certificate-based authentication has been considered keeping in mind the concurrent development of a fault-tolerant and highly available Entitlement Server. Integrating certificate-based authentication with the fault-tolerant Entitlement Server benefits the system implemented in this thesis with the advantages of a fault-tolerant server. More information about the project entitled “Fault Tolerant and Highly Available Entitlement Server” can be found in Singh’s thesis [21].

Another important consideration of this design concentrates on the security of the user data. The X.509 security certificate belonging to a user is uploaded to the server for authentication and the certificate needs to be stored in a secure environment. The certificate is stored on the server only for the time period required for authentication. As soon as the authentication procedure is complete, the certificate is deleted from the server. Finally, the design provides a simple interface for accepting user certificates and providing error messages to users in case the user authentication fails and the user is not granted access to the requested resource. The certificate authentication is designed to work together with the existing Shibboleth authentication and the user interface is similar to the existing interface to ensure uniformity throughout the authentication and authorization process.

4.2 Certificate Based Authentication

In order to understand the design of certificate-based authentication, an understanding of the Shibboleth authentication is required. Figure 2 in Chapter 2 shows the Shibboleth authentication mechanism as a part of the Shibboleth protocol. The objective of the current design of certificate-based authentication is to seamlessly replace the username/password based authentication in case the user wants to authenticate using a security certificate. The process of authentication validates the certificate and retrieves the principal name from a user certificate. The principal name is used for further authorization decisions. The principal name is a unique identifier that ties a user to the entitlements belonging to that user. The principal name of the user is the email address present in the security certificate. Figure 4 shows the protocol followed for certificate-based authentication. The design follows a client - server architecture with the client residing at the Certificate Identity Verifier and the Client Authentication Server.

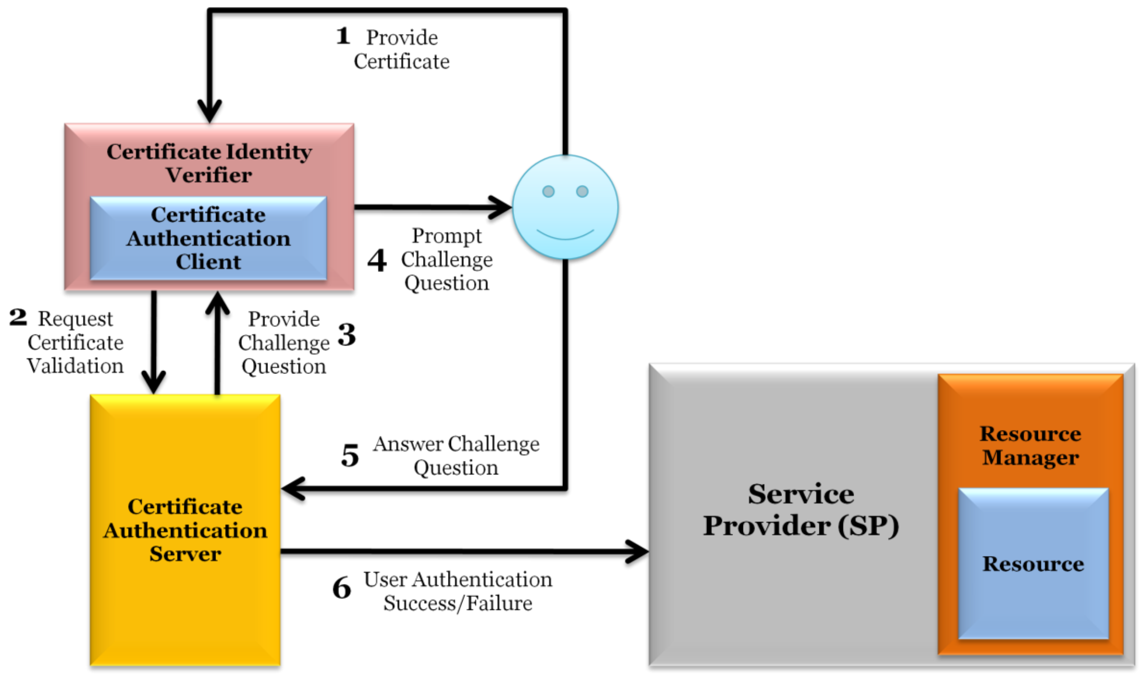

1 Provide Certificate
Certificate Identity Verifier
Certificate Authentication Client
4 Prompt Challenge Question
2 Request Certificate Validation
Provide Challenge Question 3
5 Answer Challenge Question
Certificate Authentication Server
6 User Authentication Success/Failure
Service Provider (SP)
Resource Manager
Resource

Figure 4: Certificate-based Authentication

The user provides a security certificate to the Certificate Identity Verifier for authentication. The Certificate Identity Verifier sends a message to the Certificate Authentication Server to validate the certificate. The Certificate Authentication Server extracts the principal name of the user. The server uses the OpenSSL [5] toolkit [22] to perform certificate verification and extraction of the email address that acts as a user's principal name. However, the possession of a X.509 certificate does not necessarily mean that the certificate belongs to the submitting user. The user has to answer a challenge question to confirm that the user is indeed the owner of the presented certificate. The challenge question is set by owner of the certificate and answering the challenge question is intended to prove

[5] The OpenSSL toolkit is an open-source toolkit that can be used to perform various operations on X.509 certificates like displaying certificate information and verifying the certificate chain. More information on the OpenSSL toolkit is provided in Appendix B.

that the user owns the presented certificate. Thus, after the server verifies the certificate, the Certificate Authentication Server retrieves the challenge question for the particular user and the answer to the question from the database. The principal name and the challenge question are then sent to the Certificate Authentication Client by the Certificate Authentication Server. The Certificate Identity Verifier prompts the user with the challenge question, which the user answers directly to the Certificate Authentication Server. If the user answers the question incorrectly thrice, the system provides the user with an authentication failure message and denies service to the user. In the case that user answers the question correctly, the result of the successful authentication is sent to the Service Provider to perform further authorization operations.

The decision to separate the certificate verification functionality into a server and client has been taken so that if any changes are required in the authentication mechanism, the changes are limited to the server. For example, a Virtual Organization may decide to approve certificates signed by a new root Certificate Authority. The new certificate needs to be installed on only the server housing the Certificate Authentication Server and not on all Certificate Authentication Clients in the Virtual Organization. The process of installing a new certificate authority is explained in detail in Appendix D. This server-client architecture has been designed to be similar to the architecture of the Entitlement Server so that the integration of certificate-based authentication in the Shibboleth environment can be accomplished by reusing most of the infrastructure provided by the

Entitlement Server/Client framework. The integration of certificate-based authentication with Shibboleth is explained next.

4.3 Integration with the Shibboleth Environment

The design of certificate-based authentication focuses on validating a security certificate and extracting the principal name of the user that provided the certificate. However, this authentication mechanism needs to be integrated with the Shibboleth environment and the Entitlement Server to make authorization decisions for providing access to resources provided by the Virtual Organization. Figure 5 shows the integration of certificate-based authentication into the Shibboleth environment. This mechanism is a combination of the authentication and authorization process presented in Chapter 2 and the certificate-based authentication mechanism described previously in this chapter.

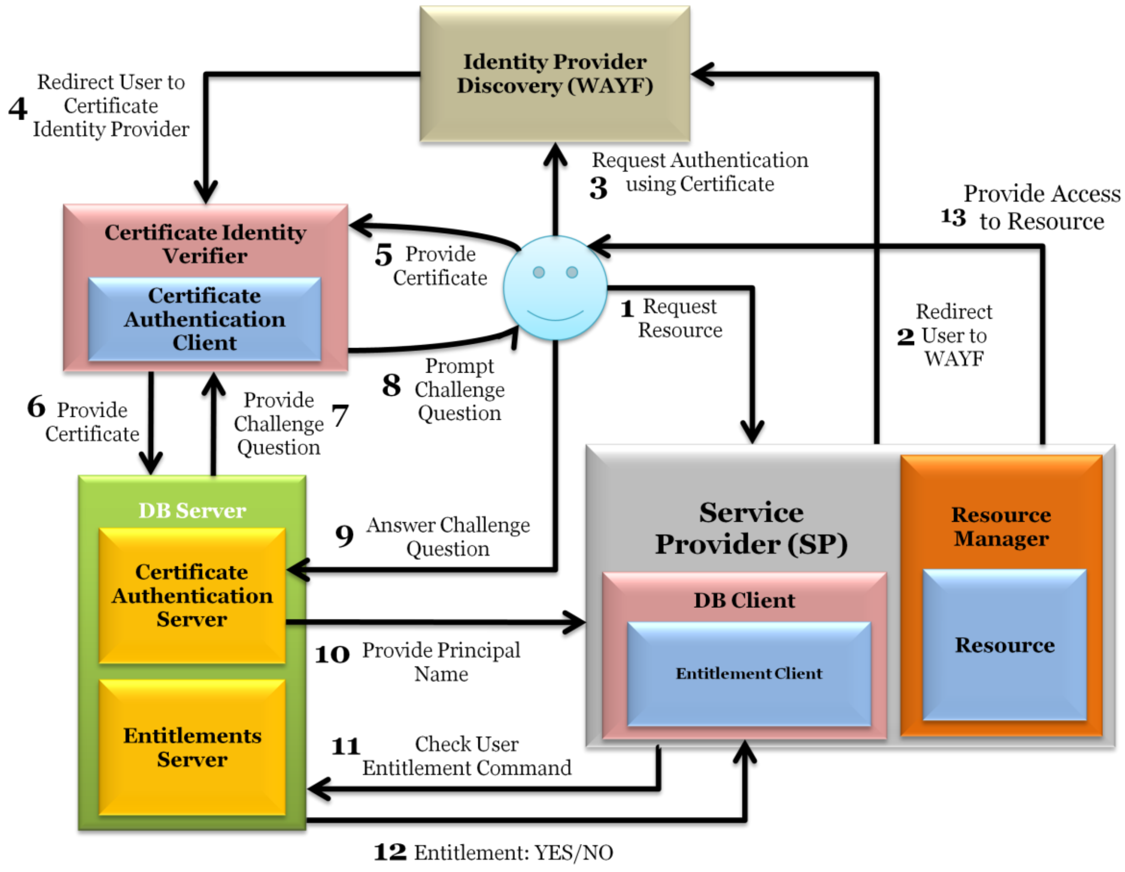

Identity Provider Discovery (WAYF)
4 Redirect User to Certificate Identity Provider
3 Request Authentication using Certificate
13 Provide Access to Resource
Certificate Identity Verifier
Certificate Authentication Client
5 Provide Certificate
1 Request Resource
2 Redirect User to WAYF
8 Prompt Challenge Question
6 Provide Certificate
7 Provide Challenge Question
DB Server
Certificate Authentication Server
Entitlements Server
9 Answer Challenge Question
Service Provider (SP)
Resource Manager
DB Client
Entitlement Client
Resource
10 Provide Principal Name
11 Check User Entitlement Command
12 Entitlement: YES/NO

Figure 5: Integration of Certificate Authentication with Shibboleth

The protocol is similar to the current authentication and authorization protocol used for Shibboleth authentication and authorization using an Entitlement Server. A user requests access to a resource provided by a Service Provider. In order to provide access to the requested resource, the user needs to be authenticated and authorized. To authenticate the user, the user is redirected to the Where Are You From (WAYF) server to determine the institution the user belongs to. However, in case of a user authenticating with a security certificate, the user requests to authenticate using a certificate instead of selecting a home institution. The WAYF directs the user to the Certificate Identity Verifier. The user provides his/her personal X.509 security certificate to the Certificate Identity Verifier which invokes the Certificate Client to verify the certificate. The certificate is transferred from the Certificate Identity Verifier to the Certificate Authentication Server to provide support for the Fault Tolerant Entitlement Server implementation. With the integration of the Fault Tolerant Entitlement Server, only a Certificate Client can communicate with the Certificate Authentication Server. The Certificate Client sends the user's certificate to the Certificate Authentication Server that verifies the certificate and extracts the principal name. The Certificate Authentication Server sends the principal name of the user and the challenge question for identity confirmation to the Certificate Client. The user is prompted with the challenge question and if the user answers the question correctly, the user is authenticated and the identity of the user is established. The principal name of the user is then sent to the Service Provider. In case the user fails to answer the question, the user is denied service as he/she is not authenticated.

Once the Service Provider has the principal name of the user providing the certificate, it can use the Entitlement Client to query the Entitlement Server and check if the user has the necessary entitlements to access the requested resource. The authorization process is the same as for Shibboleth Authentication and is not modified. Thus, a user can access protected resources using a certificate with minimal modifications to the existing system.

Figure 5 shows that the integration re-uses some of the components of the existing system. Certificate Authentication Server is a part of the Database Server and the Certificate Client reuses the infrastructure provided by a Database Client to communicate with the Certificate Authentication Server. This decision has been taken to utilize the server-client framework already available for entitlement checks. Integrating the Certificate Authentication Server with the Entitlement Server also provides certificate-based authentication with the advantages of the "Fault-Tolerant Highly Available Entitlement Server" project [21]. This project focuses on developing a fault tolerant Entitlement Server by operating multiple Entitlement Servers, as a logical group, inside a network servicing client requests. The multicast networking protocol [23] is used to achieve a distributed client service mechanism. The client application sends a query to a multicast group comprising of multiple Entitlement Servers to find an Entitlement Server that can service the client's request. The Entitlement Servers in the network stay synchronized by communicating every modification query to each other and have consistent data. Hence, any Entitlement Server in the multicast group can service the client's queries. The Fault-Tolerant Entitlement Server provides a highly available and scalable system even during failure of one or more Entitlement Servers. By coupling the Certificate Authentication Server with the Entitlement Server framework, the fault tolerant nature of the Entitlement Server is inherited by the Certificate Authentication Server. This empowers the Virtual Organization with a highly available implementation of certificate-based authentication and reduces authentication failures due to failure of one or more Certificate Authentication Servers in the network.

This chapter explained the design considerations for the implementation of X.509 certificate based authentication and incorporating it into the authentication and authorization environment. The design concentrates on modularity, data security, user-friendliness and code re-usability. The chapter proceeds to explain the protocol undertaken by the design for certificate-based authentication built upon the Shibboleth authentication mechanism. Finally, the chapter talks about how the certificate-based authentication is integrated into the existing authentication and authorization architecture and the advantages of implementing the system as per the design discussed earlier in the chapter. With a proper understanding of the system design and architecture of certificate-based authentication, the next chapter describes about the implementation of this design.

CHAPTER 5 – SYSTEM IMPLEMENTATION AND INTEGRATION WITH SHIBBOLETH

This chapter presents the implementation of certificate authentication based on the design presented in the previous chapter. The implementation of the Certificate Authentication Server, Certificate Identity Verifier and the web interface for certificate-based authentication is described. Finally, the integration of certificate authentication in the Shibboleth mechanism is discussed.

5.1 The Server Implementation

The implementation of the Certificate Authentication Server is an extension to the Entitlement Server as implemented in Ciordas's thesis [1]. Hence the Certificate Authentication Server is highly available, secure and provides a timely response for all client requests. The server runs in an endless loop accepting TCP connections from clients. The security of data transferred between the server and client is maintained by implementing a secure cryptographic design. Connections between the server and client are setup using a public key encryption mechanism and once the connection is setup, the server communicates a symmetric encryption key to the client that is used for further communication between the server and the client. Symmetric encryption proves advantageous when the connection between the server and client exists for a long duration as it is faster than asymmetric key encryption. In case of a long duration of communication, the overhead of generating a symmetric key is eventually overshadowed by the faster encryption/decryption using the symmetric key. The server uses the gdbm [24] (GNU Database Manager) database implementation on the server. Every database entry in the gdbm database has a key-value pair that is stored in a file-system based hash-table. Gdbm does not suffer from the overhead of a relational database and therefore it is extremely fast as compared to relational databases. The Entitlement Server uses gdbm because along with being fast, gdbm also provides all the functionality needed for an entitlements database. Ciordas's thesis also provides descriptions about the networking component, the cryptographic system design, the symmetric key generation and the gdbm database implementation of the server in detail [1].

In order to start the communication with the client, the server waits for the client to provide identification information and the type of operation (SP_SETUP, SP_LOOKUP, SP_USE, SP_CERT_LOOKUP) to be performed. All the operations except the SP_CERT_LOOKUP operation are handled by the Entitlement Server as these operations are specific to the authorization needs of the service provider. The SP_CERT_LOOKUP command, on the other hand, is handled by the Certificate Authentication Server and is used to verify the integrity of a user certificate, to extract authentication information and to provide a preliminary check to verify whether the user providing the certificate is indeed the owner of the certificate.

The SP_SETUP operation is used to authenticate and authorize a Service Provider from a list of known and valid Service Providers. When the Entitlement Server gets a SP_SETUP request from a Service Provider, it searches the entitlements database to check if the Service Provider has a “user” entitlement. The “user” entitlement allows a Service Provider to make entitlement queries to the Entitlement Server. If the Service Provider is known and authorized as a Service Provider, the Entitlement Server sends a “yes” response back to the Service Provider. In case of a positive response, the Entitlement Server creates a session for the Service Provider. The timestamp when the Service Provider is authenticated and authorized is saved by the Entitlement Server in order to provide service for a finite period of time. Due to this, the client must periodically re-authenticate when a session times out. This is done to enhance system security by periodically having the Service Provider prove it is alive and still the provider it claims to be, and to regenerate new symmetric keys used for encryption.

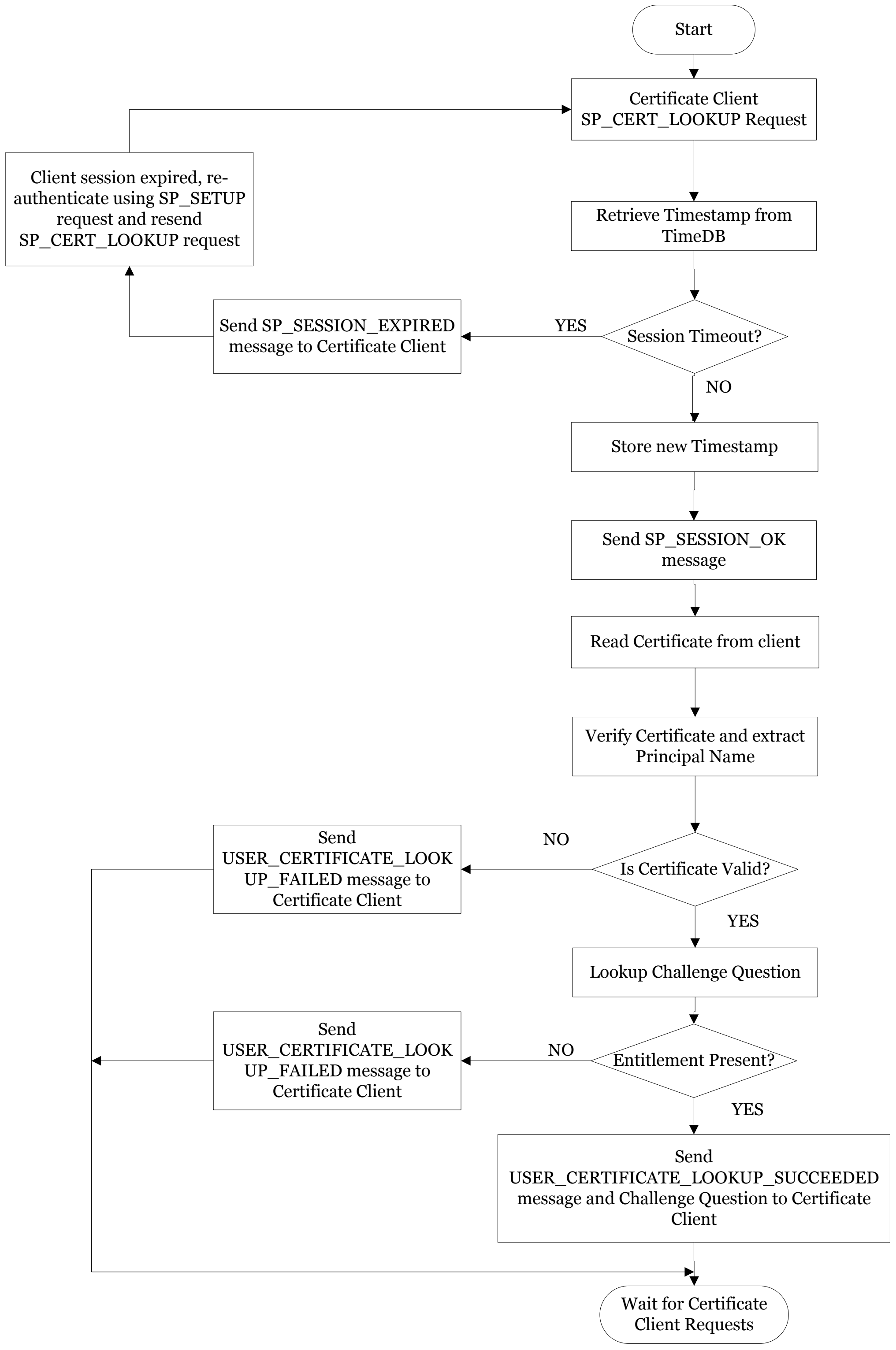
The SP_LOOKUP operation is used to lookup user entitlements from the entitlements database to help the Service Provider decide whether a user is authorized to access a particular resource. When the Entitlement Server receives a SP_LOOKUP request from a Service Provider, the Entitlement Server checks whether the Service Provider has already been authenticated in a pre-specified timeout window and has a valid session. This can be verified by checking the timestamp of authentication for the Service Provider that is present in the TimeDB database maintained by the Entitlement Server. If the authentication request of the Service Provider has timed out, the TimeDB entry for the Service Provider is deleted and the Service Provider has to re-authenticate before it can make any more entitlement requests. In case the Service Provider has a valid session, the server updates the timestamp in the TimeDB to extend the time frame for the service to the Service Provider. Once the connection has been successfully established, the Service Provider sends an entitlement query to the Entitlement Server. The Entitlement Server replies with a yes/no response that is used by the Service Provider to make authorization decisions.

The SP_USE operation is used to perform administrative operations by an authorized administrator on the entitlements database. The user requesting a SP_USE operation needs to possess an "admin" (or "root") entitlement to make changes to the entitlement database. If the user has the required entitlement, the command sent by the user to the Entitlement Server is executed and a status message is returned to the user depending on the operation requested.

In order to authenticate a user using a X.509 certificate, two stages of verification need to be performed. The first stage in authenticating a user using a X.509 certificate is verifying the certificate provided by the user. The SP_CERT_LOOKUP operation is a new operation type used when the Certificate Identity Verifier needs to verify the authenticity of a user certificate. The integration of the Certificate Authentication Server with the Entitlement Server allows the Certificate Client to reuse the SP_SETUP request, used by Service

Providers, to establish a connection with the server. The Certificate Authentication Server receiving the SP_CERT_LOOKUP request validates the Certificate Authority that signed the certificate and the validity of the certificate. The principal name of the user is then extracted from the certificate and sent back to the Certificate Identity Verifier to initiate the second stage of authentication. Since possession of a security certificate cannot be assumed to be an absolute verification of the user's identity, the second stage in authenticating a user providing the certificate is to confirm that the user is indeed the owner of the certificate. To do this, the Certificate Authentication Server returns a challenge question to the Certificate Identity Verifier. The Certificate Identity Verifier can then present the challenge question to the user and validate the answer to ensure that the user is indeed the person he/she claims to be. If the answer provided by the user, matches the preset answer to the challenge question, the user's identity is confirmed and is successfully authenticated.

Figure 6 shows the flowchart of operation of the Certificate Authentication Server after it receives a SP_CERT_LOOKUP request from a Certificate Client.



Start
Certificate Client SP_CERT_LOOKUP Request
Retrieve Timestamp from TimeDB
Session Timeout?
YES
Send SP_SESSION_EXPIRED message to Certificate Client
Client session expired, re-authenticate using SP_SETUP request and resend SP_CERT_LOOKUP request
NO
Store new Timestamp
Send SP_SESSION_OK message
Read Certificate from client
Verify Certificate and extract Principal Name
Is Certificate Valid?
NO
Send USER_CERTIFICATE_LOOKUP_FAILED message to Certificate Client
YES
Lookup Challenge Question
Entitlement Present?
NO
Send USER_CERTIFICATE_LOOKUP_FAILED message to Certificate Client
YES
Send USER_CERTIFICATE_LOOKUP_SUCCEEDED message and Challenge Question to Certificate Client
Wait for Certificate Client Requests

Figure 6: Certificate Authentication Server Operation

The Certificate Authentication Server receives the SP_CERT_LOOKUP request and the Certificate Identity Verifier's identity. The name of the Certificate Identity Verifier is searched in the TimeDB database to decide whether the Certificate Identity Verifier has been authorized to use services of the Certificate Authentication Server. As long as there are continuous additional requests from the Certificate Identity Verifier, no re-authentication needs to be done. After a timeout, a full authentication cycle must be done. Thus, if Certificate Identity Verifier's session has timed out, the Certificate Identity Verifier needs to re-authenticate using a SP_SETUP request. However, if the Certificate Identity Verifier's session is still valid, a new timestamp is stored in the TimeDB as the session timeout is extended for a new period and the Certificate Authentication Server returns a SP_SESSION_OK message to the Certificate Client. Otherwise, the Certificate Authentication Server returns an error to the Certificate Client and the request is terminated.

Upon sending the SP_SESSION_OK message to the Certificate Client, the Certificate Authentication Server receives the user's X.509 certificate over the secure connection. The certificate is temporarily stored by the Certificate Authentication Server in order to perform OpenSSL X.509 operations on the security certificate. The OpenSSL toolkit commands used to verify the certificate and to get the principal name are explained in detail in Appendix B. In case of an error, the Certificate Authentication Server returns an error code to the Certificate Client.

If the certificate verification succeeds, the Certificate Authentication Server looks for the user in the entitlement database with an entitlement “x5”. The “x5” entitlement is a special entitlement that is used to store a challenge question for confirming the identity of the user. The presence of the “x5” entitlement asserts that the user is entitled to use a security certificate for authentication. The challenge question is an additional measure of user identity security and is stored as a question followed by an answer to the challenge question in encrypted format. The challenge question and answer is unique to each user that authenticates using a X.509 certificate. If the entitlement entry is not found, an error is returned to the Service Provider. However if the entry is found, the value of the database entry is the challenge question and the SHA-2 (Secure Hashing Algorithm; see Appendix C) hash of the answer to that question. The user’s challenge question is sent to the Service Provider. The Service Provider then prompts the user with the challenge question. However, the user communicates the answer to the challenge question directly to the Certificate Authentication Server to avoid exposing the user data to the Certificate Identity Verifier. The flow chart for verification of the answer to the Challenge Question is shown in Figure 7. The Certificate Authentication Server checks if the SHA-2 hash of the answer provided by the user matches the hash value of the answer in the database. The Certificate Authentication Server communicates to the Service Provider if the user authentication was successful or not. In case of a successful certificate-based authentication, the Service Provider performs additional authorization operations.

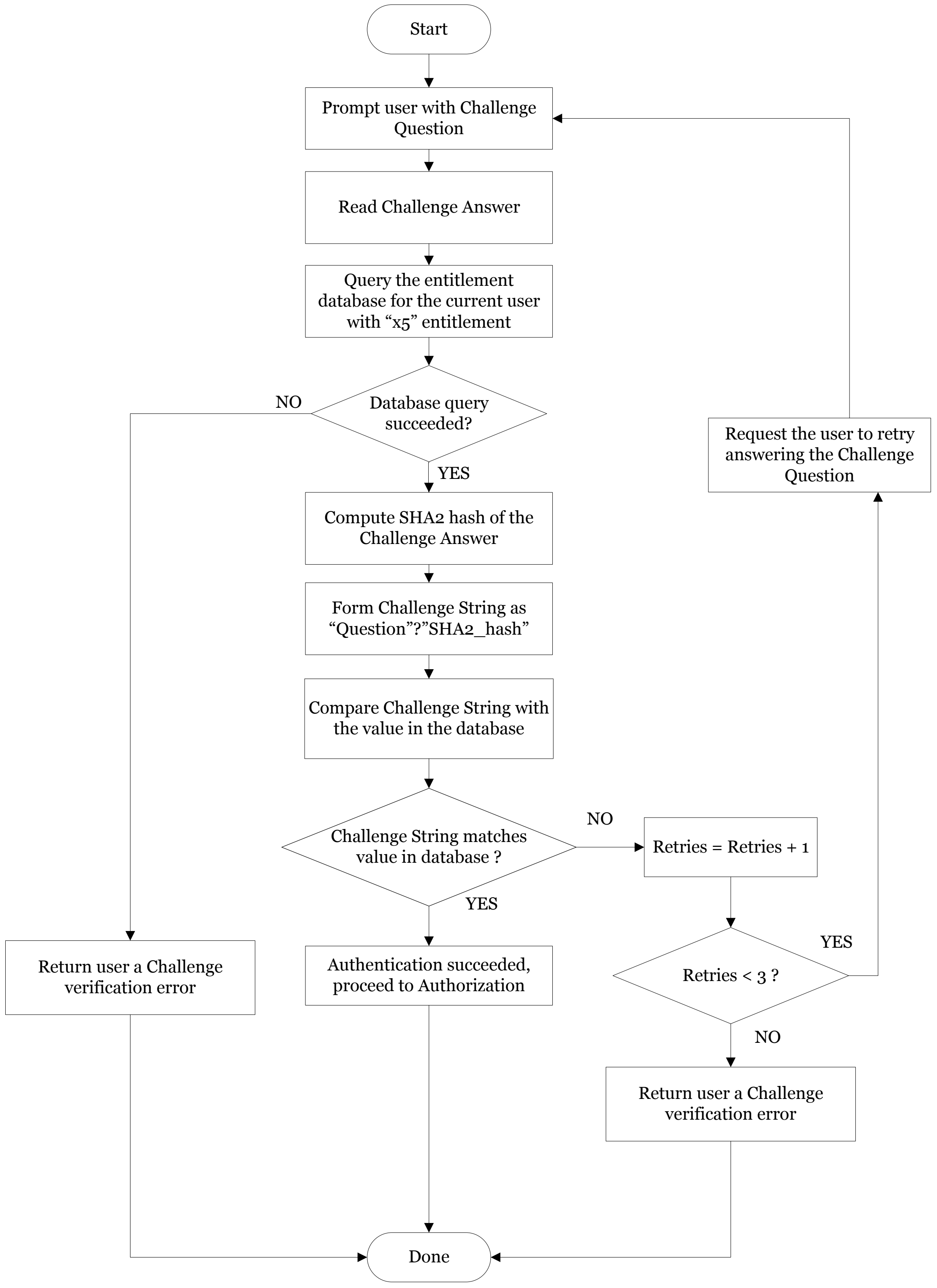
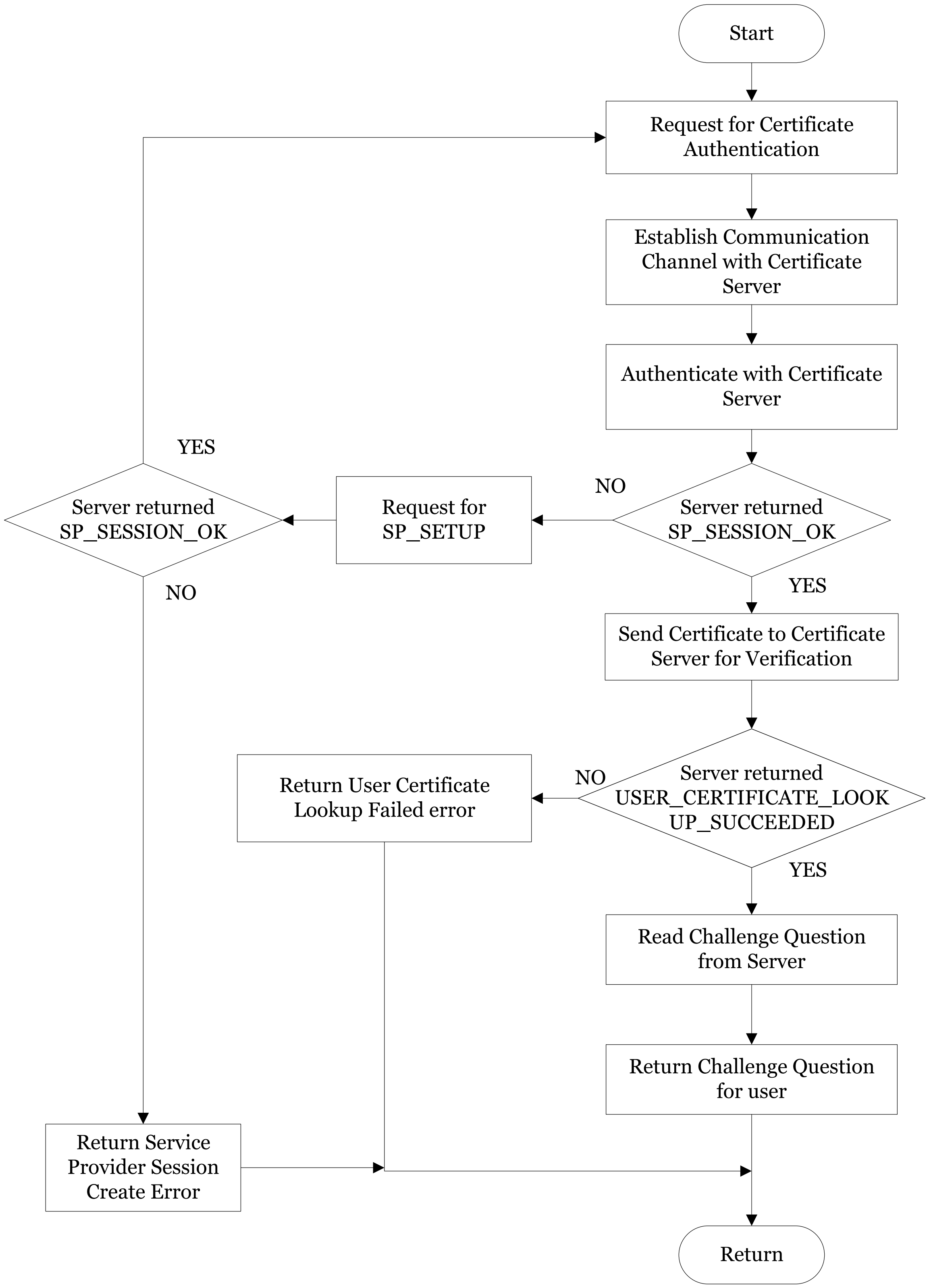

Start
Prompt user with Challenge Question
Read Challenge Answer
Query the entitlement database for the current user with “x5” entitlement
Database query succeeded?
NO
YES
Compute SHA2 hash of the Challenge Answer
Form Challenge String as “Question”?”SHA2_hash”
Compare Challenge String with the value in the database
Challenge String matches value in database ?
NO
YES
Retries = Retries + 1
Retries < 3 ?
YES
NO
Request the user to retry answering the Challenge Question
Return user a Challenge verification error
Authentication succeeded, proceed to Authorization
Return user a Challenge verification error
Done

Figure 7: Challenge Question and Answer Verification

5.2 The Client Implementation

The Certificate Client is used to make requests to the Certificate Authentication Server for authenticating certificates. The objective of the client application is to provide an interface for Certificate Identity Verifiers to use the services provided by the Certificate Authentication Server. The implementation of the Certificate Client is based on the infrastructure of the Entitlement Client. It adds an extra command for SP_CERT_LOOKUP that requests the Certificate Authentication Server to verify the user certificate and queries the user with a challenge question to confirm the identity of the user.

The Certificate Client application is supplied with a list of command line arguments that consist of the operation code for SP_CERT_LOOKUP (40), name of the Certificate Identity Verifier and the X.509 certificate filename. In order to establish communication with the Certificate Authentication Server, the Certificate Client needs to have access to the public key of the Certificate Server and the private key of the Certificate Client for the initial encryption to setup a communication channel. The initial message is encrypted using the Certificate Server's private key, which can be decrypted by the Certificate Server using its private key. However, the subsequent messages are encrypted using the client's private key and server's public key, which are decrypted by the Certificate Server using the server's private key and client's public key. This double encryption not only ensures that the message is received by the intended recipient, but also that the message was sent by the right sender.



Start
Request for Certificate Authentication
Establish Communication Channel with Certificate Server
Authenticate with Certificate Server
Server returned SP_SESSION_OK
NO
YES
Request for SP_SETUP
Server returned SP_SESSION_OK
YES
NO
Send Certificate to Certificate Server for Verification
Server returned USER_CERTIFICATE_LOOKUP_SUCCEEDED
NO
YES
Return User Certificate Lookup Failed error
Read Challenge Question from Server
Return Challenge Question for user
Return Service Provider Session Create Error
Return

Figure 8: Certificate Client Operation

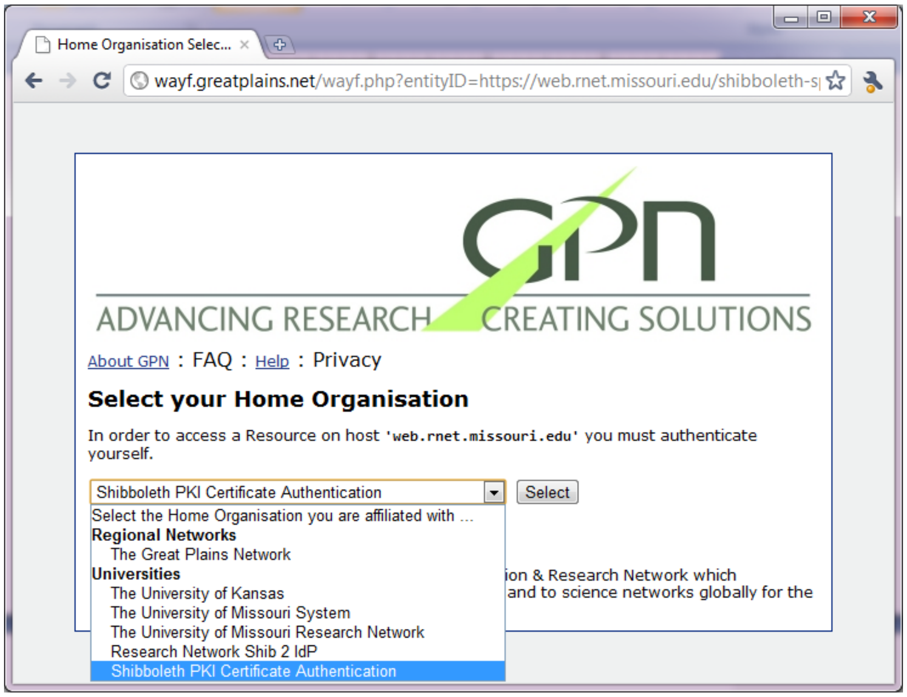
The flowchart for the Certificate Client Operation is shown in Figure 8. The client establishes a communication channel with the server and times out in case the channel cannot be established in the pre-specified timeout period. If the Certificate Identity Verifier has not been authenticated and authorized by the server, the client receives a SP_SESSION_EXPIRED message from the server. In this case the Certificate Identity Verifier has to re-authenticate before it can request any service from the server. If the Certificate Identity Verifier has a valid session with the server, it sends the user's certificate to the server. The server verifies the certificate and if the certificate is successfully verified and the principal name is successfully extracted, the client receives a USER_CERTIFICATE_LOOKUP_SUCCEEDED message from the server. In case the verification fails, a USER_CERTIFICATE_LOOKUP_FAILED message is returned to the client along with an error code that specifies the error in the verification process. After verifying the certificate, the server retrieves the challenge question for the user from its database and sends it back to the client. Upon completion of the command, the client closes the TCP connection and exits. The Certificate Identity Provider prompts the user with the challenge question to ascertain the user's identity.

Since the client code is an application, it can be invoked from the command line or through any script. In the Shibboleth-based environment, the client is invoked through PHP code on the Certificate Identity Verifier to check if the user has required entitlements to access a particular resource.

Implementation of a separate client command (SP_CERT_LOOKUP) simplifies integration of certificate authentication with the fault tolerant Entitlement Server project by separating the connection procedure from the processing of a client command. The Fault-Tolerant Highly Available Entitlement Server project implements fault tolerant behavior by running multiple servers in the network that form a logical group. A client sends a request for a connection to this multicast group. The servers that receive the request respond to the client with their individual connection details. The client chooses the first server that responded and establishes a connection after successful authentication. The procedure that follows is similar to the existing implementation of a client request. All active servers in the multicast group are synchronized and every server that responds to a client is equally capable of servicing the client's request. The Fault Tolerant Server implementation and the details of server synchronization are explained in Singh's thesis [21]. Integration of the SP_CERT_LOOKUP command provides a highly available and scalable implementation of certificate-based authentication that is capable of handling failure of one or more servers in the multicast group.

5.3 Integration with the Shibboleth Environment

Since certificate-based authentication is used in addition to username/password based authentication, changes have been made to the Shibboleth environment to provide the user an option of authenticating using a X.509 certificate. As depicted in Figure 5 in Chapter 4, when a user requests access to a particular resource to a Service Provider, the Service Provider establishes a session by directing the user to a WAYF (Where Are You From) server. The WAYF service provides the user with an additional option for authentication, namely "Shibboleth PKI Certificate Authentication" along with all the identity providers in the virtual organization as shown in Figure 9.


Home Organisation Selec...
wayf.greatplains.net/wayf.php?entityID=https://web.rnet.missouri.edu/shibboleth-s
GPN
ADVANCING RESEARCH CREATING SOLUTIONS
About GPN : FAQ : Help : Privacy
Select

Select your Home Organisation

In order to access a Resource on host 'web.rnet.missouri.edu' you must authenticate yourself.

Shibboleth PKI Certificate Authentication

Select the Home Organisation you are affiliated with ...
Regional Networks
The Great Plains Network
Universities
The University of Kansas
The University of Missouri System
The University of Missouri Research Network
Research Network Shib 2 IdP
Shibboleth PKI Certificate Authentication

ion & Research Network which
and to science networks globally for the

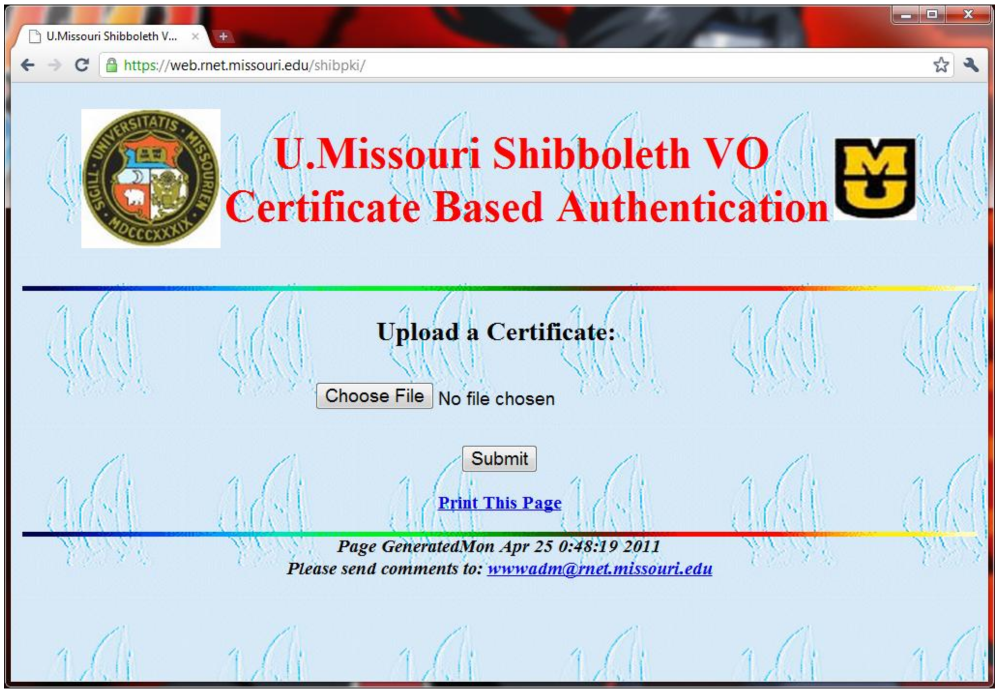
Figure 9: Option for Certificate-based Authentication

If the user selects the option of authenticating using a security certificate, the user is redirected to the authentication page on the Certificate Identity Verifier.

5.3.1 The X.509 Login Module

The Certificate Identity Verifier provides the user with an authentication page where the user can provide his/her security certificate for authentication. Such a page needs to be registered in the Shibboleth environment for use in a Virtual Organization. This is achieved by installing a X.509 login handler module with the web server [25]. The default method for login handlers is username/password based authentication. Hence the X.509 login handler is configured to accept X.509 security certificates instead of username/password combinations. The login handler sets the principal name of the user for the current Shibboleth session after successful user authentication. While the login handler prompts the user for a security certificate, it also provides the user with an option of using a security certificate that may be installed by the user in his/her web browser. The X.509 login module presents the Certificate Identity Verifier as a legitimate Identity Provider to the Shibboleth environment. Due to this login module, certificate-based authentication is integrated seamlessly in the current implementation.

The authentication page is shown in Figure 10. The authentication page accepts a X.509 certificate from a user. This page checks for the correct extension of a certificate file and the maximum file size (currently configured to 2MB) allowed to be uploaded to the server. The certificate file is then uploaded to the server, and deleted when the Certificate Identity Verifier completes the authentication process. The authentication page invokes the Certificate Client application that communicates with the server to verify the certificate provided by the user. If the certificate is authentic, the Certificate Identity Verifier needs to confirm the identity of the user providing the certificate. In order to do the confirmation, the user is presented with a challenge question as shown in Figure 11.


U.Missouri Shibboleth V...
https://web.rnet.missouri.edu/shibpki/

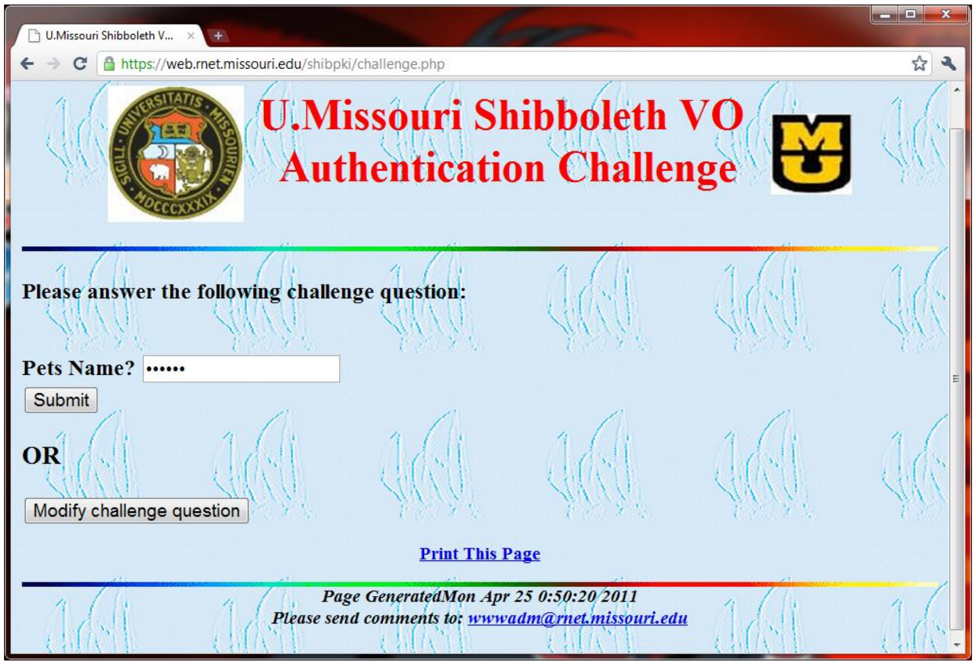
U.Missouri Shibboleth VO
Certificate Based Authentication

Upload a Certificate:

Choose File
No file chosen
Submit
Print This Page

Page GeneratedMon Apr 25 0:48:19 2011
Please send comments to: wwwadm@rnet.missouri.edu

Figure 10: Certificate-based Authentication Page



U.Missouri Shibboleth V...

https://web.rnet.missouri.edu/shibpki/challenge.php

U.Missouri Shibboleth VO Authentication Challenge

Please answer the following challenge question:

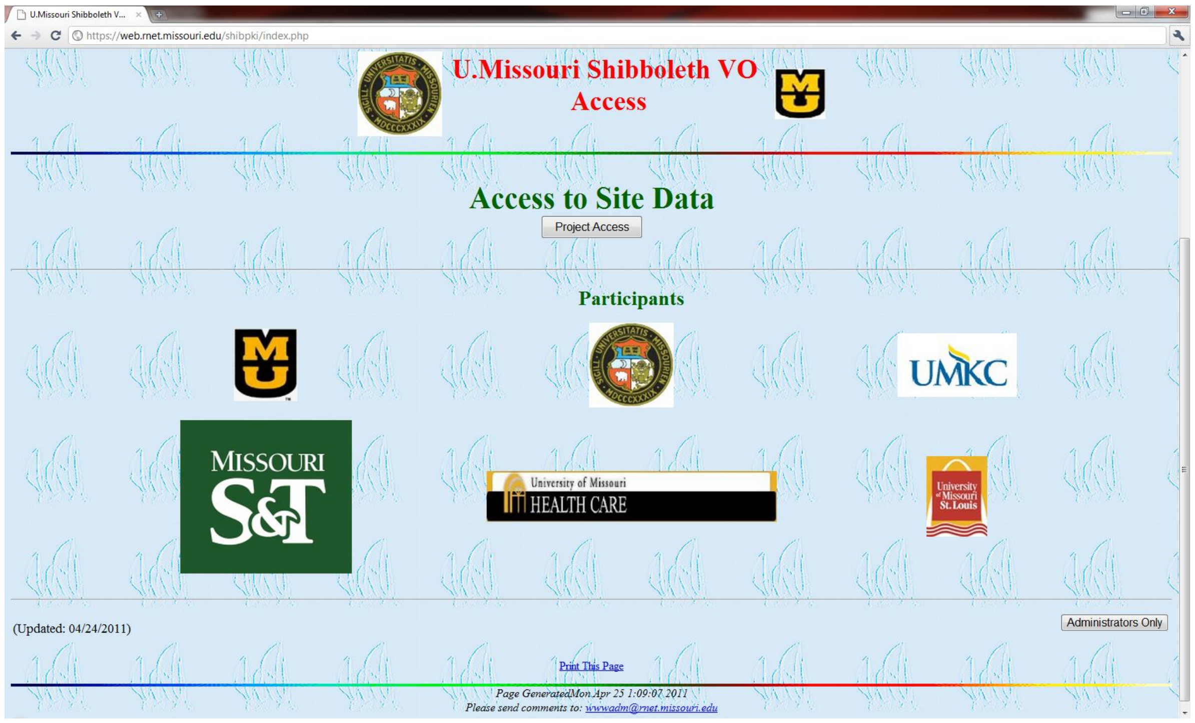
Pets Name? ••••••

OR

Page GeneratedMon Apr 25 0:50:20 2011
Please send comments to: wwwadm@rnet.missouri.edu

Figure 11: Prompt for Challenge Question

If the user answers the question correctly, the identity is confirmed and the user is directed to the requested resource and the client is invoked again to check whether the user has the necessary entitlement to access the resource. The implementation of the authorization process has not been modified. Figure 12 shows a screenshot of the user page after successful completion of the authorization process. A screenshot of the authorization failure is shown in Figure 13. An in-depth description of the authorization process can be found in Ciordas's thesis [1].


U.Missouri Shibboleth V...

https://web.rnet.missouri.edu/shibpki/index.php

U.Missouri Shibboleth VO Access

Access to Site Data

Project Access

Participants

(Updated: 04/24/2011)

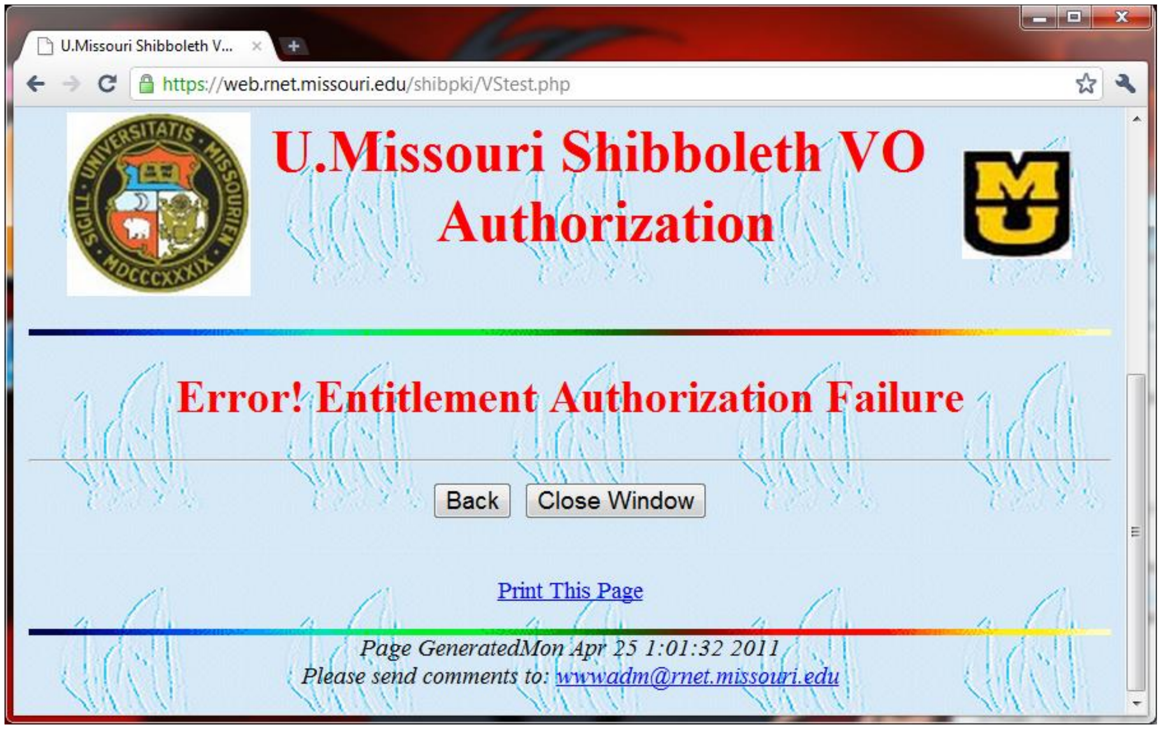
Administrators Only

Print This Page

Page GeneratedMon Apr 25 1:09:07 2011

Please send comments to: wwwadm@rnet.missouri.edu

Figure 12: User Access Granted


U.Missouri Shibboleth V...

https://web.rnet.missouri.edu/shibpki/VStest.php

U.Missouri Shibboleth VO Authorization

Error! Entitlement Authorization Failure

Back | Close Window

Print This Page

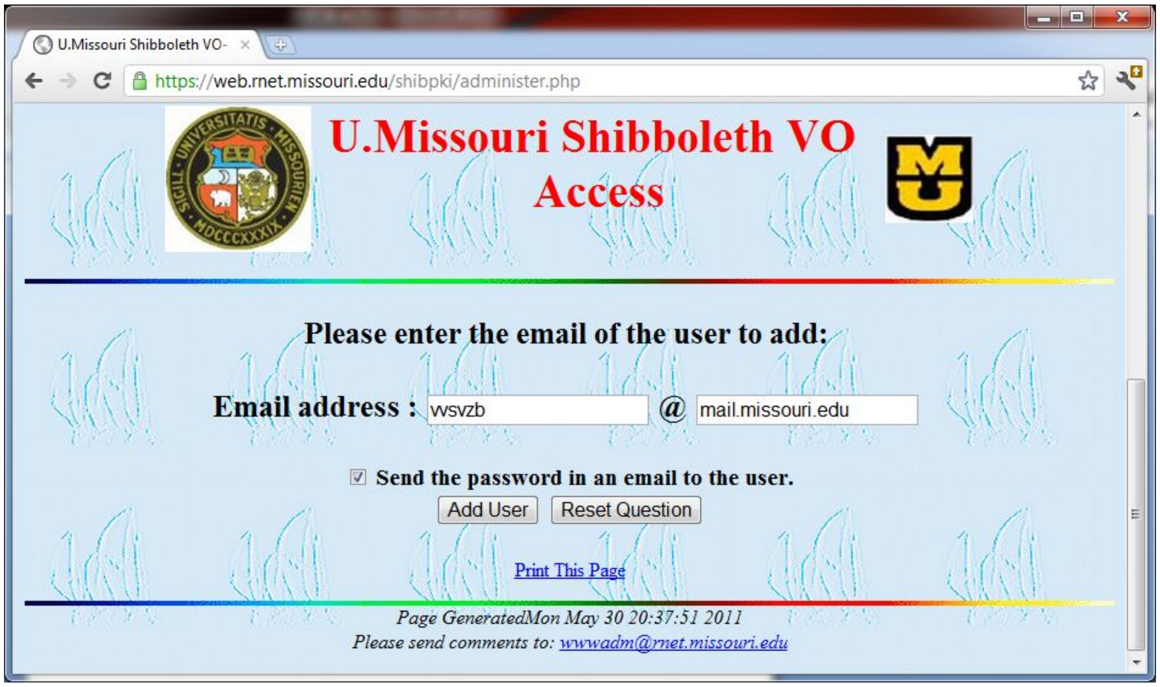
Page GeneratedMon Apr 25 1:01:32 2011

Please send comments to: wwwadm@rnet.missouri.edu

Figure 13: User Authorization Failure

5.4 *Challenge Question Administration*

The challenge question/answer mechanism is an important part of the certificate-based authentication procedure. Administrators in a Virtual Organization authorize users to use certificate-based authentication by adding users through the administrative page as shown in Figure 14. The administrators have two options, either to authorize the user to use a security certificate (Add User) or to reset the challenge question/answer (Reset Question) for the user in case the user forgets the challenge answer and wants the administrator to reset it for him/her. This procedure generates a random challenge answer for the user that is then communicated to the user. The administrator has an option of either sending the answer to the user in an email or viewing the answer on the next page.


U.Missouri Shibboleth VO-
https://web.rnet.missouri.edu/shibpki/administer.php
Add User
Reset Question
Print This Page

U.Missouri Shibboleth VO Access

Please enter the email of the user to add:

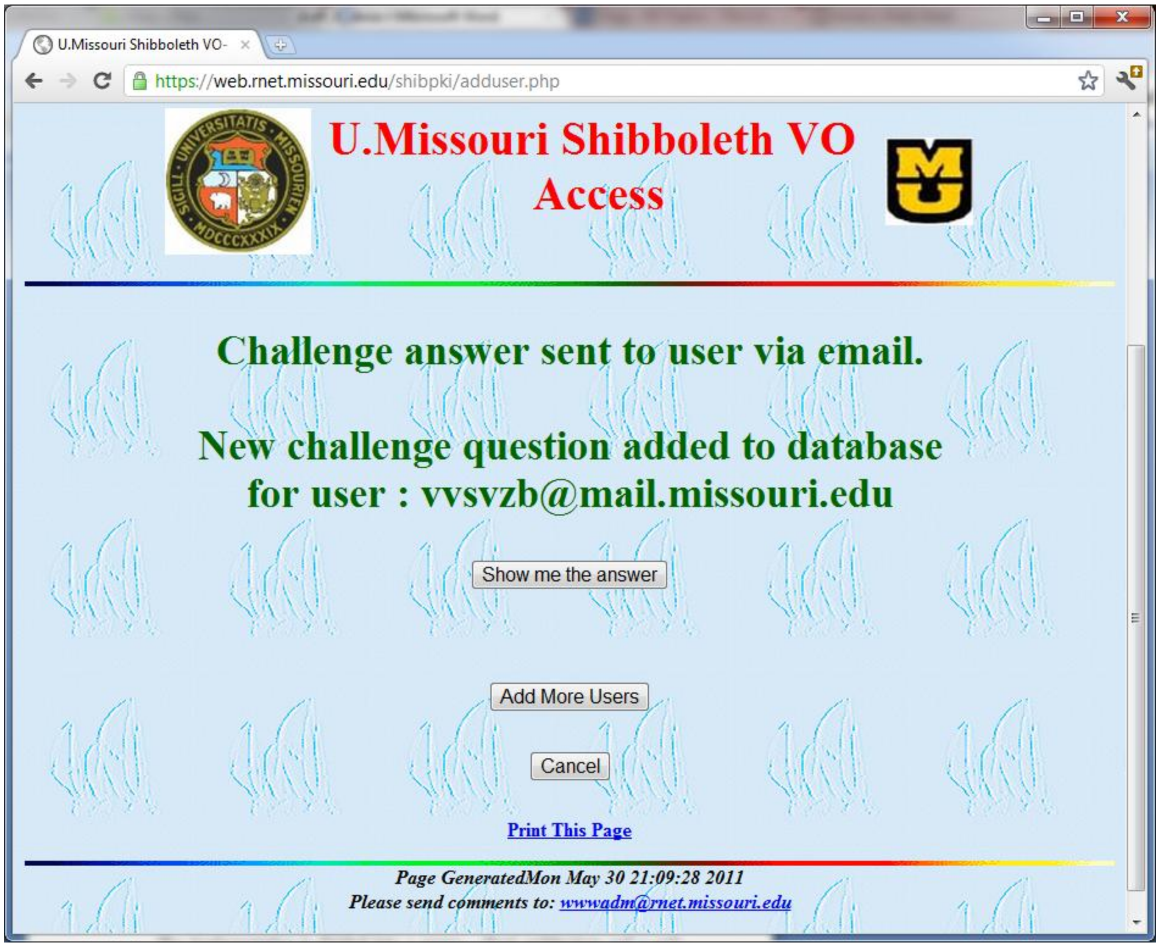
Email address : vvsvzb @ mail.missouri.edu

☑ Send the password in an email to the user.

Page GeneratedMon May 30 20:37:51 2011
Please send comments to: wwwadm@rnet.missouri.edu

Figure 14: Certificate-Based Authentication Administration

Figure 15 shows the web-page presented to the administrator after a user has been added or a user's question has been reset. The answer to the challenge question is hidden when the page is loaded and displayed only when the administrator clicks the "Show me the answer" button.


U.Missouri Shibboleth VO-
https://web.rnet.missouri.edu/shibpki/adduser.php
U.Missouri Shibboleth VO Access
Show me the answer
Add More Users
Cancel
Print This Page

Challenge answer sent to user via email.

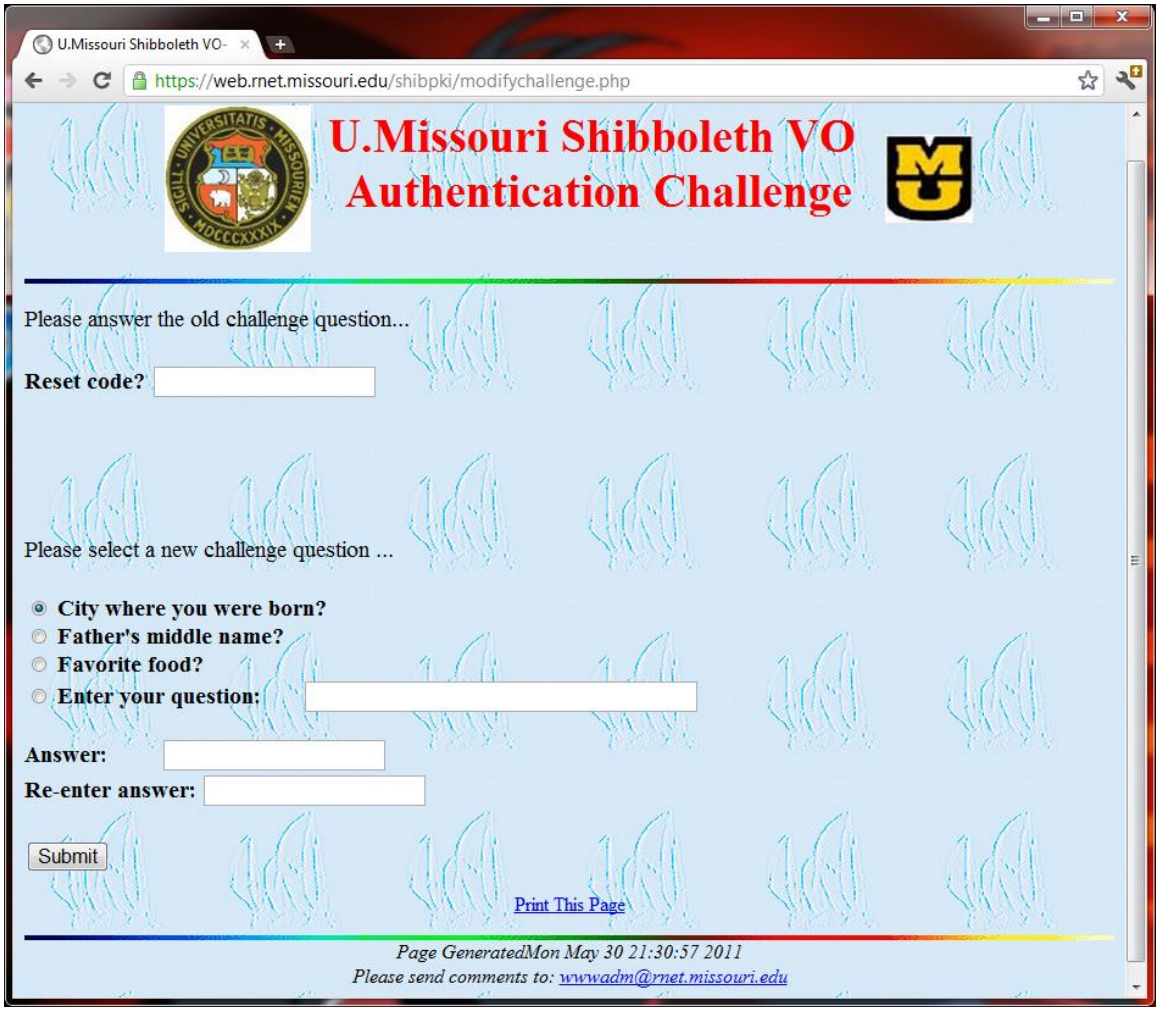
New challenge question added to database for user : vvsvzb@mail.missouri.edu

Page GeneratedMon May 30 21:09:28 2011
Please send comments to: wwwadm@rnet.missouri.edu

Figure 15: Challenge Answer Reset Page

There can be times when a user using certificate-based authentication would want to change his/her challenge question and answer. For such cases, the user can request to modify his/her challenge question using the page shown in Figure

16. The user has to first authenticate by answering the old challenge question. The user can then select a new question from the set of pre-defined challenge questions or can choose to create a new question. After submitting, the new challenge question and answer is added to the database. The user needs to answer the new challenge question for every subsequent certificate-based authentication.


U.Missouri Shibboleth VO-
https://web.rnet.missouri.edu/shibpki/modifychallenge.php

U.Missouri Shibboleth VO Authentication Challenge

Please answer the old challenge question...

Reset code?

Please select a new challenge question ...

[✓ **City where you were born?**]
Father's middle name?
Favorite food?
Enter your question:

Answer:
Re-enter answer:

Submit

Print This Page

Page GeneratedMon May 30 21:30:57 2011
Please send comments to: wwwadm@rnet.missouri.edu

Figure 16: Challenge Question Modification Page for User

This chapter explained the implementation of certificate-based authentication. The implementation is divided into a server - client architecture and a web-interface that invokes the Certificate client to communicate with the Certificate Authentication server. The server and client implement a “SP_LOOKUP_CERT” command that is used specifically to verify a user certificate and to challenge the user’s identity with a challenge question. Finally, the integration of this certificate-based authentication with the Shibboleth authentication was discussed along with screenshots of the web-pages implemented for certificate-based authentication.

CHAPTER 6 – CONCLUSION

The objective of implementing X.509 security certificate-based authentication in a Virtual Organization is to provide a way for users affiliated to institutions with or without a Shibboleth Identity Provider to securely gain access to the various institutions' resources. Authorization of users is separated by the implementation of an Entitlement Server as an extension to the Shibboleth authentication mechanism. Hence, the goal of this thesis is to provide users with an additional method of authentication in a Virtual Organization without an institutional IdP. Certificate-based authentication is an authentication method that essentially removes the dependency on a username/password based authentication.

The authentication of users using X.509 certificates brought up a few obstacles that had to be tackled during the execution of this thesis. One of the issues faced is X.509 certificate support for the various formats of X.509 certificates. The OpenSSL Application Programming Interfaces (APIs) required for validating a certificate and APIs to read the distinguished name from a certificate are different for different types of certificates. X.509 certificates are defined by the International Telecommunications Union – Telecommunications Sector standard [19]. The support for a new standard in the future would possibly require modification of the source code if older APIs are deprecated. This would require the implementation of this thesis to be maintained and kept up to date. However, updating of the implementation in this thesis is simplified by using the OpenSSL toolkit as opposed to the OpenSSL APIs. The validation of certificates and extraction of the distinguished name from a certificate is done using a “system” function call [26] that invokes the OpenSSL toolkit. The commands for certificate validation and name extraction are discussed in Appendix B. Thus, all X.509 certificates types supported by OpenSSL are inherently supported by this thesis implementation. Any modification to the X.509 certificate standard can be accommodated by simply updating the OpenSSL toolkit installed on the server(s) that hosts the Certificate Authentication Server.

Another issue faced during the implementation was to ensure that the user providing the certificate is the person that he/she claims to be. Therefore, when authenticating using a X.509 certificate, a user has to assure the system of his/her identity by answering a “Challenge Question” that can be correctly answered only by the owner of the certificate. This provides a measure of security for users authenticating using certificates and safeguards the system from unwanted users. The decision of storing the answer to the challenge question as a SHA-2 hashed value was taken after considering alternatives like storing the answer as plain-text or symmetric key encrypted values. As discussed in Appendix C, after weighing the advantages and disadvantages of all methods of storing answers to the challenge question, one-way SHA2 hashing is used to store the answer.

Certificate-based authentication opens up an array of possibilities for application in areas other than Virtual Organizations. Today, businesses are moving to online implementation and most systems require users to create a username/password based account. Maintaining many username/password combinations is a daunting task. X.509 certificates can be used in such scenarios by users to certify their identity. Users that need to authenticate can simply provide a X.509 certificate to the authentication systems as an aid to proving the user's identity. Future research in the area of certificate-based authentication can be used to achieve true single sign-on that requires the user to remember just one challenge question/answer combination instead of many username/passwords in order to access resources or services in widely distributed application domains.

In conclusion, implementation of this thesis has made computing resources accessible to a large number of users in a virtual environment that previously could not access the resources or had to acquire credentials from other institutions with a Shibboleth IdP. Certificate-based authentication is also an additional method of authentication along with username/password based authentication for current users that can authenticate at a Shibboleth IdP. The implementation of this thesis has opened up endless possibilities for organizations by demonstrating that authentication using X.509 certificates is a feasible and practical method of authentication and is comparable to username/password authentication. In fact, if certificate-based authentication is implemented at a wide-scale, just one certificate signed by a well-recognized Certificate Authority can be used to establish user identity across multiple authentication systems. Such a certificate could be used by users to authenticate at all the authentication systems that recognize the Certificate Authority. Thus, the work done in this thesis could form a basis for the concept of global single sign-on across any authentication system. This would free the users from maintaining an array of different username/passwords for every system they access, help in making computing easier and provide secure access for users.

APPENDIX A – X.509 CERTIFICATE FILE FORMAT

The format of the X.509 certificate contains the following fields.

- Version: This field specifies the version of the certificate
- Serial Number: The serial number is a unique integer value that is assigned to the certificate by the certificate issuer.
- Algorithm ID: This field specifies the identifier of the algorithm used to digitally sign the certificate.
- Issuer: The distinguished name of the Certificate Authority that signs the certificate.
- Validity: This field specifies a validity period for the certificate. It is defined by a start date and an end date.
- Subject: The subject field is used to specify the distinguished name of the owner of the certificate, i.e. the subject.
- Subject Public Key Info: This field specifies the public key along with the algorithm that is being certified by the certificate.
- Issuer Unique Identifier: This field is only in X.509 versions 2 and 3. It can be used to specify optional user about the Issuer.
- Subject Unique Identifier: Similar to the above field, this field can be used to specify optional information about the Subject.
- Extensions: This field is only in X.509 version 3. This field is used to specify some optional extensions to the certificate.

In addition to the above fields, a X.509 certificate must contain a digital signature that conforms to the algorithm specified in the Algorithm ID field. An example of a X.509 certificate is shown below.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 54061 (0xd32d)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer:   DC=org,   DC=DOEGrids,   OU=Certificate   Authorities,
CN=DOEGrids CA 1
        Validity
            Not Before: Feb 25 17:46:37 2011 GMT
            Not After : Feb 25 17:46:37 2012 GMT
        Subject:  DC=org,  DC=doegrids,  OU=People,  CN=Veerendra  Shirole
631593
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
```

```
                Modulus (2048 bit):
                    00:c4:xx:89:60:3a:80:b8:b5:3c:bc:79:c1:f9:c5:
                    2d:83:0e:db:fd:36:32:4e:d3:85:37:f0:28:d0:f2:
                    f3:3c:60:63:de:30:9c:91:49:f6:xx:72:6c:a5:5b:
                    xx:2d:e8:xx:10:69:3f:b6:51:45:bd:11:39:94:ad:
                    7e:6e:fc:b2:20:c8:38:30:15:af:dc:9f:fa:aa:f1:
                    0f:3c:7a:80:db:09:xx:8f:ea:7f:53:xx:e0:b1:80:
                    e0:fb:xx:b0:cb:2c:7f:28:f8:e2:69:89:e6:de:6e:
                    ac:39:66:ce:bf:53:fb:66:28:48:8e:de:d9:fa:xx:
                    d7:31:a4:3e:44:2f:xx:10:b4:82:9a:1d:2b:13:94:
                    17:xx:13:17:26:d5:6b:bd:8d:76:98:40:21:39:ed:
                    3a:19:25:91:52:6d:2d:7c:c6:75:85:96:8a:7d:97:
                    c6:e8:5c:06:xx:41:25:73:59:1a:xx:62:83:d5:27:
                    81:92:c9:78:05:a4:0b:ab:76:d0:5f:13:37:ec:5d:
                    92:ea:d3:6b:6c:55:67:72:14:eb:23:1f:44:1e:ad:
                    29:9f:52:81:xx:f6:c7:d7:3e:3d:c5:b8:d0:38:c1:
                    e0:f6:57:67:b2:e1:43:7c:49:d2:xx:db:29:f0:ea:
                    xx:bb:02:fb:2b:c3:xx:d8:87:28:ef:a0:52:e4:xx:
                    67:35
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Netscape Cert Type:
                SSL Client, S/MIME
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment, Data Encipherment
            X509v3 Authority Key Identifier:

keyid:CA:19:1D:12:8E:6E:FE:ED:DE:AD:B0:07:0E:08:DB:D9:8D:17:0D:5D

            X509v3 Certificate Policies:
                Policy: 1.2.840.113612.3.7.1.3.1
                Policy: 1.2.840.113612.5.2.2.1
                Policy: 1.2.840.113612.5.2.3.2.1.1

            X509v3 CRL Distribution Points:
                URI:http://crl.doegrids.org/1c3f2ca8/1c3f2ca8.crl

            X509v3 Subject Alternative Name:
                email:vvsvzb@mail.missouri.edu
    Signature Algorithm: sha1WithRSAEncryption
        a5:c2:6d:e3:58:3a:3e:4a:cd:48:cc:15:xx:25:57:43:b1:ae:
        d0:cf:91:7e:67:83:a3:b4:90:74:xx:0e:08:b6:fa:88:74:64:
        ab:d7:ab:a8:d2:xx:00:12:88:d9:07:37:cf:99:bf:ac:c6:f1:
        xx:be:e4:d0:df:7f:15:01:d2:3c:35:9f:19:xx:db:c0:08:53:
        d2:d7:10:2e:72:78:97:4e:xx:ba:c7:c7:11:a8:37:57:de:6e:
        5a:56:f2:xx:1e:0c:92:e8:19:eb:32:4c:82:f2:12:be:xx:82:
        cc:0c:d3:ce:0c:0c:68:e0:a1:99:2e:xx:64:28:db:30:e5:fe:
        f4:ba:13:9e:78:61:79:xx:8d:bc:23:64:b4:81:8e:e7:37:8c:
        f0:54:00:eb:xx:3f:99:ef:60:91:5f:cf:d1:f2:0f:28:36:dc:
        xx:c4:22:01:68:62:33:52:ec:be:ba:33:d4:xx:22:09:ce:bf:
        4a:b1:55:c8:96:cf:93:13:xx:da:64:8b:d5:f3:64:xx:ce:d4:
        a3:2a:6a:e1:31:3e:9b:46:a6:f1:8d:f1:fc:34:7a:eb:be:dc:
        74:69:1f:1c:xx:e5:5d:e9:d5:5d:9a:51:2d:f7:3f:xx:86:9c:
        a0:0e:dc:59:7b:ab:e9:73:e3:xx:08:c6:0b:eb:95:6f:7b:ef:
        5d:ef:28:66
```

This certificate is issued to the person specified in the “Subject” field as “CN=Veerendra Shirole” (CN stands for Common Name). The name of the issuer is provided in the “Issuer” field as “CN=DoeGrids CA 1”. This certificate is valid only until February 25 2012 and the user will need to request a new certificate if he wants to authenticate in the system after that date. The email field under “Subject Alternative Name” (vvsvzb@mail.missouri.edu) is considered as the “Distinguished Name” of the user. The Distinguished Name is used to uniquely identify a user in a Virtual Organization as it provides information about the principal name and the institution the user belongs to. The signature of the certificate is provided after the “Signature Algorithm” field. The signature is computed by hashing the certificate (excluding the signature itself) using the Signature Algorithm and signing it using the private key of the Issuer. This signature is used to verify the authenticity of the certificate. RSA signature algorithm and Digital Signature Algorithm (DSA) are the most frequently used signature algorithms [27]. The one-way hashing functions used for computing the hash are MD2 one-way hash function, MD5 one-way hash function and the SHA-1 hash function. A detailed description of the cryptographic algorithms used for Public Key Infrastructure can be found in [27] Section 7.

APPENDIX B – THE OPENSSL TOOLKIT

The OpenSSL toolkit is a utility that implements the Secure Sockets Layer [28] (SSL), Transport Layer Security [29] (TLS) network protocols and other cryptography standards needed for these protocols. The Certificate Authentication Server presented in this thesis uses the OpenSSL program to verify and extract information from a X.509 certificate. The OpenSSL program is a command-line tool that provides a variety of commands, each of which usually has a number of options and arguments. These commands are used for creation and management of private/public keys, X.509 certificates and Certificate Revocation Lists (CRLs). These commands are also be used for public key cryptographic operations and calculation of message digests (22). The implementation of this thesis uses the "verify" and "x509" commands of the OpenSSL program.

The "verify" command of the OpenSSL program verifies a X.509 certificate. The arguments used to verify a X.509 certificate using this command are:

```
openssl verify -CApath directory CertificateFile
```

This command verifies a certificate "*CertificateFile*" provided by the user. The option "-CApath" is used to specify a directory in which certificates of recognized Certificate Authorities (CA) are installed. Certificate Authorities are discussed in

Chapter 3. Certificate-based authentication currently supports the PEM, DER and PK12 certificate formats. The implementation of this thesis currently only authorizes DOEGrids [30] signed certificates for authentication. The program returns a success message if the certificate is valid. In case the certificate verification fails, the program provides an error message, such as in case of an expired certificate the program provides the following message:

```
CertificateFile: /DC=org/DC=doegrids/OU=People/CN=Optimus Prime
310002
error 10 at 0 depth lookup:certificate has expired
```

If a certificate is successfully validated, the email address of the user is extracted to get the identity of the person providing the certificate. The email address is used to uniquely identify the user as a principal name as well as the institution the user belongs to. To extract the email address, the “x509” command of the OpenSSL program is invoked. The “x509” command is used for X.509 Certificate Data Management in OpenSSL. The “x509” command can be used to display certificate information, convert certificates to various forms or edit certificate trust settings. The “x509” command, like the “verify” command, provides some options to perform specific operations on the certificate. In order to extract the email address of the user from the certificate, the “x509” command of the openssl program is used with the following parameters.

```
openssl x509 -noout -in CertificateFile –email
```

The “-noout” option is used to disable the printing of the encoded version of the openssl request. The “–in” option requests OpenSSL to read the certificate “CertificateFile” as the input certificate. The “-email” option is used to output the email address that is present in the certificate. The email address is extracted by reading from the stdout when the OpenSSL command is executed. The email address is then used to determine the principal name of the user and the institution the user belongs to.

The return value of the OpenSSL commands can be used to determine the cause of a failure, in case there is one. These error codes are propagated to the user to provide a description of the cause for failure. Certificate related errors occur if certificate is of a type not recognized by OpenSSL, the certificate is signed by a Certificate Authority not authorized by the Virtual Organization or if the certificate has expired. In case the certificate does not contain an email address field, it is also considered as an error as without an email address the user identity cannot be successfully established in the system. Thus the OpenSSL program is used to verify a user X.509 certificate to ascertain the validity of the certificate provided by the user and to obtain the identity of the user.

APPENDIX C – SECURE HASHING ALGORITHM (SHA-2)

The Secure Hashing Algorithm (SHA-2) is used in certificate-based authentication to compute the hash values of answers to the challenge questions. SHA-2 is a family of hash functions comprising of four standard functions that are SHA-224, SHA-256, SHA-384 and SHA-512 [31]. SHA-256 and SHA-512 are the more popular functions from the SHA-2 set of functions. While the SHA-512 is more secure than the SHA-256 function, the current implementation of certificate-based authentication uses the SHA-256 standard for generating a hash value. This is because SHA-256 suffices the requirements of this thesis, uses less memory and is faster than the SHA-512 function. The SHA-256 hashing algorithm takes an input message of length $< 2^{64}$ bits and generates a message digest that is the hash value of the input message. A detailed description of SHA-2 and its implementation can be found in [31].

Secure Hashing Algorithms are one-way functions that make it computationally infeasible to find the message from the message digest. As the size of the message digest is always constant, it is harder to even predict the size of the message in case of hashing as compared to other symmetric encryption methods. The SHA-2 algorithm is therefore used as a one way encryption mechanism to store the hash values of answers to challenge questions in certificate-based authentication.

Storing answers as hash values instead of plain-text answers protect the answers from users who have access to the database where the answers are stored. In case the answers are stored as plain-text, anyone with access to the database would be able to read and use the challenge answers. To generate a message with the same digest as another message is also computationally infeasible. Thus it is impracticable for someone to access the database of answer hashes and try to generate a message with the same hash as that of a challenge answer. This one-way encryption is also more effective than symmetric encryption of challenge answers, as if the encryption key for symmetric encryption is compromised all the answers in the database would be compromised. However the SHA algorithms do suffer from some shortcomings such as in case a user forgets his/her challenge answer, there is no way of recovering the answer from the database. A new answer will have to be created by the administrator and communicated to the user. In spite of this drawback, hashing still proves to be a secure way of storing private user data and is very hard to compromise. Therefore SHA-2 hash values of the challenge answer are stored into the entitlement database to verify with the user answer when the user answers a challenge question.

APPENDIX D – CERTIFICATE AUTHORITY INSTALLATION

With the incorporation of certificate-based authentication, a Virtual Organization can decide to approve certificates signed by a new Certificate Authority (CA). If such a scenario arises, the certificate of the new Certificate Authority needs to be installed on the server hosting the Certificate Authentication Server. The process of a new Certificate Authority installation is explained in this section.

As discussed in Appendix B, the OpenSSL command to validate a certificate accepts the name of a Certificate Authority certificates directory as a command-line argument. This directory (for example /usr/local/ssl/certs) is where all the CA certificates of CAs' acceptable to the Virtual Organization are installed. In order to install a new Certificate Authority, a PEM [32] (Privacy Enhanced Mail) formatted certificate of the Certificate Authority is required. Certificates from any other format need to be converted to a PEM format certificate. The CA certificate is copied into the CA certificates directory. However, during validation of a certificate, OpenSSL tries to locate a CA certificate in the CA certificates directory using the CA certificate's 8-byte hash value as a symbolic link for the CA's certificate. This hash value is calculated by using the OpenSSL command:

```
openssl x509 -noout -hash -in ca-certificate-file
```

Thus a symbolic link that points to the original CA certificate needs to be created and given the name as the hash value. For example if the output of the hash generation command is “**df11c5fd**”, the symbolic link to the CA’s certificate file is given the name “df11c5fd.0”. The “.0” extension is given to the symbolic link only if the hash for this CA certificate is unique. In case a symbolic link with the name “df11c5fd.0” is already present, the symbolic link to the new CA certificate is named as “df11c5fd.1” and so on. An example of the contents of the CA certificate directory is shown next:

```
[vshirole@web certs]$ ls -l *.0
-rw-r--r-- 1 xxxxxxx xxx 1436 Oct 26  2006 1c3f2ca8.0
-rw-r--r-- 1 xxxxxxx xxx 1448 Oct 26  2006 d1b603c3.0

[vshirole@web certs]$ openssl x509 -noout -in d1b603c3.0 -subject
subject= /DC=net/DC=ES/O=ESnet/OU=Certificate Authorities/CN=ESnet Root CA 1

[vshirole@web doegrids]$ openssl x509 -noout -in 1c3f2ca8.0 -subject
subject= /DC=org/DC=DOEGrids/OU=Certificate Authorities/CN=DOEGrids CA 1
```

There are two CA certificates installed in this folder. The name of the first CA is “ESnet Root CA 1” and it is a local Certificate Authority. The second CA is the “DOEGrids CA” which is a well recognized Certificate Authority. Any certificate signed by either of these Certificate Authorities will be accepted by a system using this CA certificates directory. Additional CA certificates can be added to this directory as symbolic links as explained previously. Once this process is completed, the new CA certificate is successfully installed. Any subsequent commands to verify a user certificate will use all the CA certificates in the CA certificate directory, including the newly installed CA certificate, to validate a user certificate. By installing multiple CA certificates, the Virtual Organization can support certificates signed by various Certificate Authorities that are acceptable to the Virtual Organization.

APPENDIX E –DEBUG OUTPUT FOR CERTIFICATE-BASED AUTHENTICATION

The –v option of the Certificate Authentication Server application is used to enable debugging output for certificate-based authentication. When the debugging output is enabled, the Certificate Authentication logs additional information and error messages to a log file ("cert_test.log") in the current working directory. In case of errors with authentication process, the log file can be used to determine causes of failure. An example of the contents of the log file can be given as.

```
***** New Certificate Authentication Request *****
Mon May 30 17:10:29 CDT 2011
User Certificate = phpF5DPvC.pem, User VO = umsystem.edu
------ Extracting Username ------
ClarkKent@cs.missouri.edu
Extracting user email from certificate - SUCCESSFUL !!!!!

------ Verifying the certificate -------
Distinguished Name:  Clark Kent 790192
Email Address:  ClarkKent@cs.missouri.edu
Certificate Issued By:  DOEGrids CA 1
***Return Value:  0
Certificate verification - SUCCESSFUL.

------ Extracting challenge question -------
Challenge question found.

Challenge question answered incorrectly.
User Authentication using Certificate = FAILED.
--------------------------------------------
***** New Certificate Authentication Request *****
Mon May 30 17:12:20 CDT 2011
User Certificate = phpycTx2o.pem, User VO = umsystem.edu
------ Extracting Username ------
PeterParker@missouri.edu

Extracting user email from certificate - SUCCESSFUL !!!!!
```

```
------ Verifying the certificate -------
Distinguished Name:  Peter Parker 16366
Email Address:  PeterParker@missouri.edu
Certificate Issued By:  DOEGrids CA 1
***Return Value:  0
Certificate verification - SUCCESSFUL.

------ Extracting challenge question -------
Error!   Challenge   Absent.   Cert   =   9562phpycTx2o.pem,   Email   =
PeterParker@missouri.edu
Mon May 30 17:12:21 CDT 2011
Action: Need to authorize user to use a certificate.

Error !!!! Challenge question - ABSENT.
--------------------------------------------
***** New Certificate Authentication Request *****
Mon May 30 17:12:56 CDT 2011
User Certificate = phpT3L7fM.pem, User VO = umsystem.edu
------ Extracting Username ------
Error! Email Absent. Cert = 9991phpT3L7fM.pem, User VO = umsystem.edu
Mon May 30 17:12:56 CDT 2011
Action: Provide a certificate in an acceptable format with a valid email
address.
Extracting user email from certificate - FAILED !!!!!

---------------------------------
Error! Email Absent. Cert = 9991phpT3L7fM.pem, User VO = umsystem.edu
***** New Certificate Authentication Request *****
Mon May 30 17:14:51 CDT 2011
User Certificate = phprYp0ut.pem, User VO = umsystem.edu
------ Extracting Username ------
ClarkKent@cs.missouri.edu
Extracting user email from certificate - SUCCESSFUL !!!!!

------ Verifying the certificate -------
Distinguished Name:  Clark Kent 790192
Email Address:  ClarkKent@cs.missouri.edu
Certificate Issued By:  DOEGrids CA 1
***Return Value:  0
Certificate verification - SUCCESSFUL.

------ Extracting challenge question -------
Challenge question found.
Challenge question answered correctly!
User Authentication using Certificate = SUCCESSFUL.
```

The log file provides a step by step log of the actions taken during the certificate-based authentication process of a user. In case the authentication fails, the log file also provides the action that needs to be taken for successful authentication. The contents of the log file can be searched for the keyword “Error” to determine all the failures that occurred since the log file was last cleared. Similarly, other keywords like “Challenge Absent” or “Certificate Format” can be used to search for specific errors. An example of one such particular error (“Challenge Absent”) is provided next:

```
[vshirole@web client]$ grep "Challenge Absent" -n cert_test.log
40:Error! Challenge Absent. Cert = 9562phpycTx2o.pem, Email =
PeterParker@missouri.edu
97:Error! Challenge Absent. Cert = 10213phpiWylKE.pem, Email =
PeterParker@missouri.edu
```

The line provides information about the name of the temporary certificate file, the username extracted from the certificate and the reason for authentication failure. The line number of the failure can be used to view the file and get detailed information about the steps taken during authentication of the particular certificate. Thus the debug output can helpful for administrators in locating errors faced by users in the authentication process.

BIBLIOGRAPHY

1. Ciordas, Ionut Ovidiu. Fine-grained authorization in the Great Plains Network virtual orgranization. *M.S. Thesis, Computer Science Department.* University of Missouri, August 2007. https://mospace.umsystem.edu/xmlui/handle/10355/4967

2. Great Plains Network. *http://www.greatplains.net/.* [Online] [Cited: July 03, 2010.] http://www.greatplains.net/.

3. Shibboleth. *http://shibboleth.internet2.edu/.* [Online] [Cited: July 02, 2010.] http://shibboleth.internet2.edu/.

4. Morgan, R. L., et al. Federated Security: The Shibboleth Approach. *EDUCAUSE Quarterly.* 2004, Vol. 27, 4 p12-17.

5. ITU-T. Recommendation X.509: The Directory - Authentication Framework. *International Telecommunication Union.* [Online] 1988. [Cited: September 08, 2010.] http://www.itu.int/rec/T-REC-X.509-198811-S/en.

6. Weise, Joel. Public Key Infrastructure Overview. *Sun BluePrints Online.* August 2001. http://www.sun.com/blueprints/0801/publickey.pdf.

7. Opplinger, Rolf. *SSL and TLS: Theory and Practice.* Artech House, 2009.

8. Mowshowitz, Abbe. Virtual organization: A vision of management in the information age. *The Information Society.* 1994, Vol. 10, 4, pp. 267-288.

9. The Open Source Definition. *Open Source Initiative.* [Online] [Cited: July 24, 2010.] http://www.opensource.org/docs/osd.

10. Single Sign-On. *Open Group.* [Online] [Cited: July 24, 2010.] http://www.opengroup.org/security/sso/.

11. Internet2 Middleware Initiative. *www.internet2.edu.* [Online] [Cited: July 24, 2010.] http://www.internet2.edu/middleware/index.cfm.

12. Security Assertion Markup Language. *Cover Pages.* [Online] [Cited: July 12, 2010.] http://xml.coverpages.org/saml.html.

13. Shibboleth Goals. *https://spaces.internet2.edu.* [Online] [Cited: July 17, 2010.] https://spaces.internet2.edu/display/SHIB2/USGoalReq.

14. Understanding Shibboleth. *https://spaces.internet2.edu.* [Online] [Cited: July 17, 2010.] https://spaces.internet2.edu/display/SHIB2/NewUnderstandingShibboleth.

15. SPPKIConfig. *https://wiki.shibboleth.net.* [Online] [Cited: April 3, 2011.] https://wiki.shibboleth.net/confluence/display/SHIB/SPPKIConfig.

16. Shirley, R. Internet Security Glossary. [Online] 2000. [Cited: September 08, 2010.] http://www.ietf.org/rfc/rfc2828.txt.

17. Katz, Jonathan and Lindell, Yehuda. *Introduction to Modern Cryptography.* CRC Press, 2007.

18. Rivest, R. The MD5 Message-Digest Algorithm. [Online] 1992. [Cited: September 08, 2010.] http://tools.ietf.org/html/rfc1321.

19. International Telecommunications Union - Telecommunication Standardization Sector. *http://www.itu.int.* [Online] [Cited: September 07, 2010.] http://www.itu.int/net/ITU-T/info/Default.aspx.

20. Certificate Revocation and Status Checking. *http://technet.microsoft.com.* [Online] [Cited: September 07, 2010.] http://technet.microsoft.com/en-us/library/bb457027.aspx.

21. Singh, Harcharan. Fault Tolerant and Highly Available Entitlement Server. *M.S. Thesis, Computer Science Department.* University of Missouri, Aug 2011

22. About the OpenSSL Project. *www.openssl.org.* [Online] [Cited: October 01, 2010.] http://www.openssl.org/about/.

23. Introduction To Multicast. *http://www.firewall.cx.* [Online] [Cited: April 17, 2011.] http://www.firewall.cx/multicast-intro.php.

24. gdbm. *www.gnu.org.* [Online] [Cited: October 04, 2010.] http://www.gnu.org/software/gdbm/.

25. X.509 Login Handler. *https://wiki.shibboleth.net/.* [Online] [Cited: May 18, 2011.] https://wiki.shibboleth.net/confluence/display/SHIB2/X.509+Login+Handler.

26. Linux Man Pages. *www.linux.com.* [Online] [Cited: December 20, 2010.] http://www.linux.com/learn/docs/man/4262-system3.

27. R. Housley, W. Ford,W. Polk,D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. *http://tools.ietf.org.* [Online] January 1999. [Cited: May 18, 2011.] http://tools.ietf.org/html/rfc2459.

28. Weaver, A.C. Secure Sockets Layer. *Computer*. 2006, Vol. 39, 4.

29. Dierks, T. and Rescorla, E. The Transport Layer Security (TLS) Protocol. [Online] April 2006. [Cited: October 01, 2010.] http://tools.ietf.org/html/rfc4346.

30. DOEGrids Certificate Service. *http://www.doegrids.org*. [Online] [Cited: May 18, 2011.] http://www.doegrids.org/index.html.

31. Secure Hash Standard. [Online] August 2002. [Cited: December 04, 2010.] http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf.

32. Kent, S. Privacy Enhancement for Internet Electronic Mail. [Online] February 1993. [Cited: March 02, 2011.] http://tools.ietf.org/html/rfc1422.

33. OpenSSL. *www.openssl.org*. [Online] [Cited: 10 12, 2010.] http://www.openssl.org/docs/apps/openssl.html.