NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SINGLE SIGN-ON SOLUTION FOR MYSEA SERVICES

by

Sonia Bui

September 2005

Thesis Advisor:	Cynthia E. Irvine
Co-Advisor:	Thuy D. Nguyen

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Single Sign-on Solution for MYSEA Services			**5. FUNDING NUMBERS**
6. AUTHOR(S) Sonia Bui			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The Monterey Security Architecture (MYSEA) is a trusted distributed environment enforcing multilevel security policies. To provide a scaleable architecture, a federation of MYSEA servers handles service requests. However, the introduction of multiple servers creates security and usability problems associated with multiple user logins. A single sign-on solution for the MYSEA server federation is needed. After user authenticates once to a single MYSEA server, the user's credentials are used to sign on to the other MYSEA servers.

The goal of this thesis is to create a high-level design and specification of a single sign-on framework for MYSEA. This has entailed a review and comparison of existing single sign-on architectures and solutions, a study of the current MYSEA design, the development of a new architecture for single sign-on, an analysis of single sign-on threats within a MYSEA context, a derivation of single sign-on objectives in MYSEA, leading up to the security requirements for single sign-on in MYSEA. Security and functionality are the main driving factors in the design. Others factors include performance, reliability, and the feasibility of integration into the existing MYSEA MLS network. These results will serve as a basis for a detailed design and future development of sign-on in MYSEA.

14. SUBJECT TERMS information assurance**,** single sign-on, distributed authentication, Monterey Security Architecture, multilevel security, federation, Common Criteria			**15. NUMBER OF PAGES** 106
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

SINGLE SIGN-ON SOLUTION FOR MYSEA SERVICES

Sonia Bui
Civilian, Naval Postgraduate School
B.S., Santa Clara University, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2005

Author: Sonia Bui

Approved by: Cynthia E. Irvine, Ph.D.
Thesis Advisor

Thuy D. Nguyen
Co-Advisor

Peter J. Denning, Ph.D.
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Monterey Security Architecture (MYSEA) is a trusted distributed environment enforcing multilevel security policies. To provide a scaleable architecture, a federation of MYSEA servers handles service requests. However, the introduction of multiple servers creates security and usability problems associated with multiple user logins. A single sign-on solution for the MYSEA server federation is needed. After user authenticates once to a single MYSEA server, the user's credentials are used to sign on to the other MYSEA servers.

The goal of this thesis is to create a high-level design and specification of a single sign-on framework for MYSEA. This has entailed a review and comparison of existing single sign-on architectures and solutions, a study of the current MYSEA design, the development of a new architecture for single sign-on, an analysis of single sign-on threats within a MYSEA context, a derivation of single sign-on objectives in MYSEA, leading up to the security requirements for single sign-on in MYSEA. Security and functionality are the main driving factors in the design. Others factors include performance, reliability, and the feasibility of integration into the existing MYSEA MLS network. These results will serve as a basis for a detailed design and future development of sign-on in MYSEA.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
- **A. MOTIVATION....1**
- **B. PURPOSE OF STUDY....2**
- **C. ORGANIZATION OF THESIS....2**

II. BACKGROUND....3
- **A. INTRODUCTION....3**
- **B. MONTEREY SECURITY ARCHITECTURE....3**
  - **1. MYSEA Servers....3**
  - **2. MYSEA Clients....4**
  - **3. Single Level Networks....4**
  - **4. Dynamic Security Services....5**
- **C. SINGLE SIGN-ON....5**
  - **1. Overview of SSO Architectures....6**
  - **2. Comparison of SSO Architectures....8**
  - **3. Conclusions of SSO Study....14**
- **D. SUMMARY....14**

III. MYSEA SSO REQUIREMENTS ANALYSIS....15
- **A. INTRODUCTION....15**
- **B. SYSTEM ARCHITECTURE....15**
  - **1. System-Supported Services....15**
  - **2. System Components....16**
- **C. CONCEPT OF OPERATIONS....18**
  - **1. Initial User Authentication....18**
  - **2. Single Sign-On Process....21**
  - **3. User Session Status Update Process....23**
  - **4. Failure Recovery....25**
- **D. SINGLE SIGN-ON SYSTEM DESCRIPTION....27**
- **E. ASSUMPTIONS....28**
- **F. THREAT ANALYSIS....29**
- **G. ORGANIZATIONAL SECURITY POLICIES....31**
- **H. OBJECTIVES....32**
  - **1. Security Objectives for the System....32**
  - **2. Security Objectives for the Environment....34**
- **I. SYSTEM LEVEL REQUIREMENTS....35**
  - **1. Single Sign-on Requirements....35**
    - ***a. Authentication Server Requirements....35***
    - ***b. Trusted Path Extension Requirements....36***
    - ***c. Application Management Server Requirements....37***
    - ***d. Administrator Requirements....37***
    - ***e. User Requirements....37***

2. Service Management Requirements....38
a. Authentication Server Requirements....38
b. Trusted Path Extension Requirements....38
c. Application Management Server Requirements....38
d. Administrator Requirements....39
J. SUMMARY....39
IV. SECURITY REQUIREMENTS....41
A. INTRODUCTION....41
B. AUTHENTICATION SERVER SECURITY FUNCTIONAL REQUIREMENTS....41
1. Authentication Server Audit....41
2. Authentication Server Communication....42
3. Authentication Server Cryptography....42
4. Authentication Server Data Protection....42
5. Authentication Server Identification and Authentication....43
6. Authentication Server Protection....43
7. Authentication Server Resource Management....44
8. Authentication Server Security Management....44
9. Authentication Server Access....45
10. Authentication Server Trusted Path/Channels....45
11. Authentication Server Single Sign-on Management....45
C. AUTHENTICATION SERVER SECURITY ASSURANCE REQUIREMENTS....46
1. Authentication Server Configuration Management....46
2. Authentication Server Operation....47
3. Authentication Server Development....47
4 Authentication Server Guidance Documents....47
5 Authentication Server Life Cycle Support....48
6 Authentication Server Test Coverage....48
7 Authentication Server Vulnerability Assessment....48
D. THREAT AND POLICY MAPPING....48
E. ASSUMPTION MAPPING....67
F. REQUIREMENTS MAPPING....68
G. SUMMARY....78
V. FUTURE WORK AND CONCLUSIONS....79
A. INTRODUCTION....79
B. FUTURE WORK....79
1. Additional Requirements....79
2. Prospective Design Work....80
C. CONCLUSIONS....81
LIST OF REFERENCES....83
INITIAL DISTRIBUTION LIST....85

LIST OF FIGURES

Figure 1. New MYSEA System Architecture 18
Figure 2. User Authentication Process 19
Figure 3. Selecting the Run Option 20
Figure 4. Changing the Session Level 21
Figure 5. User Accessing an AMS Application 22

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Comparison of various SSO architectures 12
Table 2. Comparison of SSO architectures (cont.) 13
Table 3. System Assumptions 29
Table 4. System Threats 31
Table 5. Organizational Security Policies 31
Table 6. System Security Objectives 34
Table 7. Operational Environment Security Objectives 35
Table 8. Threat to Objective Mapping 63
Table 9. Policy to Objective Mapping 67
Table 10. Assumption to Environmental Objective Mapping 68
Table 11. Objectives to Requirements Mapping 77

THIS PAGE INTENTIONALLY LEFT BLANK

ABBREVIATIONS AND ACRONYMS

AMDB	Application Mapping Database
AMS	Application Management Server
AUS	Authentication Server
CA	Certificate Authority
CC	Common Criteria
CIM	Consistency Instruction Manual
CM	Configuration Management
CONOPS	Concept of Operations
DAC	Discretionary Access Control
DSS	Dynamic Security Services
FIPS	Federal Information Processing Standards
ID	Identifier
IT	Information Technology
LAN	Local Area Network
MAC	Mandatory Access Control
MLS	Multilevel Security
MYSEA	Monterey Security Architecture
NIST	National Institute of Standards and Technology
NSA	National Security Agency
PC	Personal Computer
PCC	Protected Communications Channel
PKI	Public Key Infrastructure

SAK	Secure Attention Key
SSO	Single Sign-on
SSS	Secure Session Server
STOP	Secure Trusted Operating System
TCB	Trusted Computing Base
TCS	Trusted Channel Server
TOE	Target of Evaluation
TPE	Trusted Path Extension
TPS	Trusted Path Server
TSF	Target of Evaluation Security Functions

ACKNOWLEDGMENTS

I thank my advisors Dr. Cynthia Irvine and Thuy Nguyen for their guidance, knowledge, and support. I have learned so much from our synergistic discussions. I thank David Shifflett and Jean Khosalim for sharing their expertise with me in the MYSEA project. I thank Tim Levin for guiding me early on in this project.

I thank my husband for his loving support during my graduate studies. I especially thank my mother for all the sacrifices she has made for me throughout the years.

This material is based upon work supported by the National Science Foundation under Grant No. DUE-0114018. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

Currently, IT systems must be able to support numerous services and applications to accomplish the mission of the enterprise. For improved performance and reliability, these services and applications may be distributed across multiple machines in the enterprise network. As with a stand-alone machine, users must also authenticate to these networked machines in order to access the applications hosted by them. If no system-wide authentication architecture exists, the user may be forced to enter authentication information multiple times, at least once for each network application used. The inconvenience of multiple authentications not only causes users to lose productivity, but also imposes more administrative overhead in managing the machines to ensure the enterprise security policy is enforced by all machines.

Single sign-on (SSO) has been hailed as a solution to deal with the usability and security problems associated with multiple user authentications. Single sign-on provides users the convenience of authenticating once to access applications hosted on multiple machines. In addition, SSO provides the enterprise the ability to centralize authentication administration and management. This helps to ensure that the security policy is consistently enforced throughout the organization. SSO has been used extensively in business and academic environments.

The need for single sign-on extends to environments enforcing multilevel security (MLS), such as those found in the military and intelligence communities. The Monterey Security Architecture (MYSEA) is one such environment. MYSEA is a trusted distributed networking environment enforcing multi-domain security policies. Currently, MYSEA is designed to support a single server hosting several of applications and a limited number of clients that make application requests to the server. In order to accommodate a growing number of MYSEA clients requiring access to a wider variety of services, a federation of MYSEA servers is used to handle the service requests [1].

The introduction of multiple servers into MYSEA introduces security and usability problems (as previously described) caused by the user having to sign on

multiple times. A single sign-on solution for the MYSEA federation is needed to offset these problems. A user would only need to authenticate once to a single MYSEA server; the user's credentials can then be used to sign in to the other MYSEA servers that the user needs to access.

B. PURPOSE OF STUDY

The goal of this thesis is to create a high-level design and specification of a single sign-on framework for MYSEA. Accomplishing this goal involves a review and comparison of existing single sign-on architectures and solutions, and a study of the current MYSEA design. To incorporate SSO capabilities into the MYSEA environment, a new MYSEA architecture and a concept of operations for SSO needs to be developed. Ultimately, security requirements for SSO support in MYSEA will be specified based on an analysis of single sign-on threats within a MYSEA context, the environmental assumptions, the organizational policies, and the single sign-on objectives for MYSEA.

Security and functionality are the main driving factors in the design. Other factors considered are performance, reliability, and the feasibility of integration into the existing MYSEA MLS network. The results of this thesis will serve as a basis for future design refinements and future development of protocols to facilitate single sign-on in MYSEA.

C. ORGANIZATION OF THESIS

This thesis is organized into five chapters. Chapter I first introduces the need for single sign-on in the MYSEA project. Chapter II provides a background on MYSEA, an overview and comparison of current SSO architectures, and an analysis of integrating SSO into MYSEA. Chapter III provides the requirements analysis for MYSEA support of SSO capabilities. This chapter includes a high-level design of the new system architecture and concept of operations for SSO support. The threats, assumptions, organizational policies, objectives, and system-level requirements for the SSO system are also analyzed in Chapter III. Chapter IV lays out the initial security functional and assurance requirements for the SSO system based on the analysis provided in Chapter III. Finally, Chapter V discusses the future work and conclusions for this thesis.

II. BACKGROUND

A. INTRODUCTION

This chapter will provide a background on the Monterey Security Architecture (MYSEA) and a brief analysis of existing single sign-on architectures applicable to a similar architectural context.

B. MONTEREY SECURITY ARCHITECTURE

MYSEA [1] provides a distributed networking environment for the enforcement of mandatory security policies. It consists of many low assurance commercial products and a small but sufficient number of high assurance elements. This architecture includes the following components and services: the high assurance MYSEA servers, low assurance MYSEA clients, legacy single level networks, and dynamic security services.

1. MYSEA Servers

Each high assurance MYSEA server consists of a DigitalNet XTS-400 Trusted Computer System running the DigitalNet Secure Trusted Operating System (STOP). The STOP kernel is the foundation for the system's Trusted Computing Base (TCB) which enforces a MLS policy using mandatory access control (MAC) and discretionary access control (DAC). The MAC of the STOP complies with the rules in the Bell-LaPadula model for information secrecy [2] and the rules in the Biba model [3] for information integrity. Both these models have been proven to be complete and secure with respect to confidentiality (the Bell-LaPadula model) and integrity (the Biba model).

Work on the MYSEA project extended the XTS-400 functionality to include a multilevel Trusted Path Server (TPS), a Secure Session Server (SSS), and the Trusted Channel Server (TCS). The TPS component creates a trusted path to a remote MYSEA client through which identification and authentication, security session level negotiation, password modification, and other trusted path services are performed. The TPS component stores the user session information in the User Database. The User Database contains tuples that specify a unique username, the TPE ID associated with the user, the

status of the session, the security level of the session, and any other security pertinent information. The TPS component is responsible for creating, modifying, and deleting the entries in the User Database.

The SSS process is used to launch untrusted application services (such as web servers or mail servers) running at the same security level as the MYSEA client requesting the service. The SSS component determines the security level of the requesting client by querying the User Database.

More information on the TPS, SSS, User Database and MYSEA server-client interaction can be found in [4]. The TCS will be discussed later in the section on single level networks.

2. MYSEA Clients

Each MYSEA client consists of an untrusted commercial-off-the-shelf personal computer and a Trusted Path Extension (TPE). These PCs run popular commercial operating systems and software applications that are familiar to users. MYSEA clients support users of different security levels, so they need to address the object reuse requirement – no residual data pertaining to a subject, logged in at a specific security level, remains on the PC after that subject has logged out. MYSEA PCs are therefore thin and stateless; on every user session login, the operating system and all applications are loaded from a non-writeable source into volatile memory. Upon user logout, all information in the volatile memory is purged, ensuring no remnants of the previous user's information remains.

Each TPE is a physical device that is both physically and logically associated with a MYSEA client. Juxtaposed between the client and the LAN, the TPE creates a trusted path between user and the remote MYSEA server. It facilitates the user login process, allowing the user to authenticate to the server and set the security level of the current session. Only then is the PC-based client allowed to connect to the local area network and access application services on the MYSEA servers.

3. Single Level Networks

MYSEA can also interface with pre-existing single level networks. Through the introduction of a Trusted Channel Server (TCS), each network, operating at a single

security level, can communicate with the multilevel MYSEA server that hosts the TCS. The TCS is similar to the TPE in that the TCS creates a secure, unforgeable link between a single level network and a MYSEA server. The TCS is responsible for managing the protocol used in the initiation and termination of each trusted channel in the Protected Communications Channel, the channel used for all MLS LAN communications. The TCS is also responsible for associating a sensitivity level for each inbound network connection to the MYSEA server and checking the sensitivity level on each outbound connection from the MYSEA server. For information on the design of the TCS, see [5].

4. Dynamic Security Services

The MYSEA server also provides Dynamic Security Services (DSS) for the MYSEA environment. DSS is analogous to the Quality of Services concept in networking – DSS is focused on quality of the security service instead of network service. DSS allows for the level of security assurances to be modulated based on the environmental conditions. For example, due to high computational load sensitive but unclassified data may be encrypted with a moderate cryptographic algorithm instead of a stronger algorithm (which could be used when the computational load is lighter). A prototype incorporating DSS-enabled capabilities with IPsec (Internet Protocol Security) was created for use in MYSEA; see [6] for more details on this project and more information on DSS.

C. SINGLE SIGN-ON

Authentication is the process by which a computer system confirms the identity of an individual, usually based on a name and password. Single sign-on (SSO) is a specialized form of authentication that allows a user to authenticate once in a particular system and thereafter gain access to multiple systems and services. Single sign-on relieves the burden on the user of having to enter authentication information multiple times (e.g., once for every service accessed). In addition, single sign-on facilitates the application of a consistent authentication policy across a domain based on centralized management of authentication [7].

Numerous single sign-on solutions have been developed by industry and academia. SSO solutions can be organized into two main categories [7]: those that deal with a single set of credentials, and those that deal with multiple sets of credentials. The

difference between the two categories is the number of user credentials handled by the SSO solution in a deployment environment. A SSO solution dealing with a single set of credentials only has to handle one type of authentication credential per user; for example, one common authentication mechanism is a username and password, so in SSO all the systems in the domain generally support the same authentication mechanism and accept the same password for an individual user. SSO solutions that handle multiple sets of user credentials usually operate across separate domains that may each require a separate credential. For the scope of this thesis, only SSO solutions that manage one credential per user will be studied because the same credential is recognized by all MYSEA servers in a particular domain. More information on SSO solutions that deal with multiple sets of credentials, such as a list of existing commercial solutions (such as Passgo SSO), can be found in [7].

SSO solutions that handle a single set of credentials can be further categorized based on its SSO architecture. The architectural categories that were examined for this thesis were: authentication database replication, token-based SSO, public key infrastructure-based SSO, proxy-based SSO, and identity-provider redirection. A description of each of these SSO architectures will be provided in the following section.

1. Overview of SSO Architectures

The simplest SSO architecture is authentication database replication. Clients authenticate to a central authentication server and the server stores information about currently logged in clients in a session database. This database is then broadcasted to all the servers. In essence, the authentication server serves as the “master” holder of the authentication database and all the other servers are the “slaves”. When a client contacts another server to request a service, the server authenticates the client based on its copy of the authentication database and allows the client to connect if the client is found in the replicated database.

In token-based SSO, a client authenticates to an authentication server and gets back from the server a cryptographic token. The client uses the token to prove its identity to each application server it wants to access. The server does some cryptographic processing on the token to verify the identity of the client and validity of the token. Tokens rely on shared secret keys and represent the trust between the application server

and the authentication server. The classic example of a token-based SSO is the Kerberos authentication protocol which involves additional tokens (called "tickets" in Kerberos) and additional client-server messages for single sign-on [8].

Public key infrastructure (PKI)-based SSO requires that users register themselves to a certification authority (CA). The registration process involves users proving their identities with credentials, the generation of private key - public key pairs, and the creation of user certificates (which contains the public key for the respective user) by the CA. The client uses the private key (which it only knows) and the certificate issued by the CA to generate tokens (similar to those in token-based SSO) that are used for authentication and SSO. The main differences between PKI-based SSO and token-based SSO are the user registration process in PKI and the use of asymmetric cryptography in PKI vs. the use of symmetric cryptography in token-based SSO. There is currently no PKI-based SSO standard, but many PKI-based SSO solutions have been developed by both academia and industry. An interesting solution that combines token-based authentication of Kerberos and PKI is SESAME (Secure European System for Applications in a Multivendor Environment). SESAME also has an option that only uses a PKI [9].

In a proxy-based SSO, the user authenticates to the central authentication server, and the authentication server itself supplies the user credential (e.g., username and password) to the appropriate server whenever the user requests to use an application on another server. Proxy-based SSO is used often when servers have different authentication mechanisms and the user has to have multiple sets of credentials. The authentication proxy server uses a database to maintain all the credentials for the user. However, proxy-based SSO can still be used even when there is one set of credentials per user; the database on the proxy would simply maintain a single set of credentials for each user. Proxy-based SSO solutions are popular since they do not require much modification to the end systems to enable single sign-on. One example of a commercial proxy-based SSO product is Novell Nsure SecureLogin [10].

Identity-provider redirection SSO is used mainly over the Internet to allow users to access resources on websites located in different domains. When an unauthenticated

user's web browser requests a resource from a site, the site redirects them to an identity provider. The user then authenticates to the identity provider and the identity provider returns authentication information, such as the user's password or a ticket, to the user's browser, commonly as a browser cookie. The user's browser is then redirected back to the initial site (with the resource the user wanted) and presents the authentication information to the site. The site examines the authentication information and allows the user to access the resource based on the information. Microsoft's Passport is probably the most well known example of identity-provider redirection SSO [11].

The next section will compare these different SSO architectures across a set of factors.

2. Comparison of SSO Architectures

The SSO architectures introduced in the previous section were examined and compared to each other based on the following criteria: performance bottlenecks, scalability mechanisms, implementation requirements, consistency issues, potential security problems, and security benefits.

SSO architectures were analyzed for the existence for any potential performance bottlenecks. SSO performance bottlenecks are the places where network traffic is forced to go through a single point during a SSO session, or they may be places where the throughput (network or computational) may be slow. These performance bottlenecks are of interest because they can indicate possible failure points or slow points and may affect decisions on resource allocation (more resources should be devoted to these bottlenecks to improve performance and security). For authentication replication SSO architectures, the performance bottleneck is the frequency of the authentication information updates from the master authentication server to the slave servers because the slaves depend entirely on the master server to supply them the latest authentication information. The performance bottleneck for token-based SSO is at the central authentication server because the client must ask the authentication server for a token for each separate service. PKI-based SSO has a performance bottleneck in the checking of expired or revoked certificates, a performance factor for PKI in general. The proxy authentication server is the obvious bottleneck for proxy-based SSO since it handles all authentication communications between the client and various servers. For identity-provider redirection

SSO, the potential bottleneck is at the identity-provider since all unauthenticated clients are redirected to it by all the sites that refer to it for client authentication.

Scalability mechanisms are those that facilitate the expansion of the number of clients that a SSO solution can support. This implies that there must be a way to increase the number of authentication servers in order to handle an increased number of clients. In an authentication information replication SSO architecture, there could be multiple authentication servers to handle the load, but these servers will still need to be slaves to a master authentication server in charge of all the authentication updates. A token-based SSO can also use replicated authentication servers, but there must be a single master authentication server or some other consistency mechanism. A PKI-based SSO architecture is scalable through the use of certificate chaining, allowing a CA to verify certificates issued by other CAs. A proxy-based SSO can increase the number of proxies to handle a larger client load. In identity-provider redirection, the identity provider can provide multiple servers to distribute the user authentication workload. Again, there are consistency issues in the proxy-based and identity-provider based SSO with the use of multiple authentication servers.

Another factor examined were the implementation requirements for specific SSO architectures. For authentication replication, the authentication servers and all the other servers have to be able to recognize the same authentication format since no additional software is being used to support SSO. Token-based SSO requires clients and servers to recognize and use tokens; for example, in Kerberos, applications have to be "kerberized" before the application can be used for SSO. PKI-based SSO also have a similar requirement in that clients and servers have to support the use of certificates. Implementation requirements for proxy-based SSO are almost minimal, for clients and application servers need little, if any, modification – almost all the SSO complexity is pushed to the authentication proxy. Identity-provider redirection SSO also does not require complex clients (just standard web browsers), but does require services to be web-based.

Consistency issues were also investigated for the different SSO architectures. The most important thing that needs to be consistent is the authentication database containing

the currently authenticated client information. Any SSO architecture that may employ multiple authentication servers (authentication information replication, token-based, proxy-based, and identity-provider redirection SSOs) requires a coherent authentication database to ensure correct access control of users to the resource or application – the distributed information must be managed such that the effective authentication policy is the same as provided by a single authentication database. Token-based SSO also may have expiration dates on the tokens, so the expired tokens have to be consistently denied by all servers, and the use of timestamps in some token-based SSO solutions (like Kerberos) require synchronized clocks. PKI-based SSO has consistency issues with the revocation of certificates, which, like tokens, need to be uniformly denied by every server.

Another factor examined was the security benefits of adopting a particular SSO architecture. Authentication replication SSO provides increased availability because authentication information is stored in multiple places (slave servers), so if the authentication server were to fail, a slave server may be able to take its place. Increased availability can be a security benefit for the other SSO architectures that can employ multiple authentication servers (token-based, PKI-based, proxy-based, and redirection to identity-provider SSOs). The timestamps in token-based SSO provide protection against replay attacks in which an attacker presents a previously used token. PKI-based SSO allows for the mutual authentication of the client to the server and server to the client through the examination of both server and client certificates. The security benefits in a proxy-based or identity-provider redirection SSO architecture depends on the security mechanisms enforced at the proxy or identity provider since either of them can incorporate tokens or certificates in the authentication process.

Potential security problems for these SSO architectures were also explored. Again, in architectures where any authentication replication takes place, the use of outdated authentication information by a server can lead to unauthorized user access. It is also obvious that the compromise of an authentication server for all of these architectures would be disastrous as any secret or private keys may be exposed. Potential security problems particular to token-based SSOs usually involve client-side caching of the tokens, since these tokens may be stolen or reused if the clients are not secure.

Token-based SSOs may also use timestamps, requiring all clocks to be synchronized securely to avoid servers synchronizing to a rogue server's clock. Both proxy-based and identity-provider redirection SSO may be susceptible to man-in-the-middle attacks in which clients expose their passwords to an attacker's machine posing as the proxy, application server, or the identity provider. Identity-provider redirection SSO also has a serious problem of clients not having to authenticate before requesting a service. In the other SSO architectures, clients are forced to authenticate before attempting to contact an application server.

Table 1 summarizes the observations made for the various SSO architectures in regards to performance bottlenecks, scalability mechanisms, and implementation requirements. Table 2 summarizes the observations based on consistency issues, security benefits, and potential security problems.

The evaluation of whether a particular SSO architecture is "better" than another, and the choice of a using particular SSO architecture, can only be made in context of the environment and organization that will use SSO. For this thesis, the environment is that used in MYSEA, and the next section will include an evaluation and choice of SSO architecture.

SSO Architecture	Performance Bottlenecks	Scalability Mechanisms	Implementation Requirements
Authentication information replication	Frequency of authentication database updates	Addition of more slave servers	All servers need to understand the authentication format
Token-based	Authentication server needs to be contacted for each application request	Replicated authentication servers	Additional code required at application servers and clients
PKI-based	Checking of revoked certificates	Chaining of certificates from different CAs	Requires support of certificates by servers & clients
Proxy-based	Proxy handles all client-server communications	Addition of more proxies	Thin clients, unmodified application servers, complexity pushed to proxies
Identity-provider redirection	Authentication at the identity provider	Replication of authentication servers at the identity provider	Web-based services, standard web browsers for clients

Table 1. Comparison of various SSO architectures

SSO Architecture	Consistency Issues	Security Benefits	Potential Security Problems
Authentication information replication	All slave copies need to be consistent	Increased availability of authentication information	Outdated authentication info in slaves may lead to unauthorized access
Token-based	Expired tokens, clock synchronization, consistency among replicated authentication servers	Timestamps prevent replay attacks	Client caching of tickets, authentication server compromise disastrous, time synchronization issues
PKI-based	Certificates need to be current and valid	Mutual authentication of client and server	Rogue CAs; revoked certificates possibly used
Proxy-based	Proxies should be consistent	Depends on the authentication mechanism enforced at proxy	Compromised proxies or fake proxies
Identity-provider redirection	Consistency among replicated authentication servers	Depends on the authentication mechanism enforced at identity provider	Man-in-middle attacks, clients not authenticated before service request

Table 2. Comparison of SSO architectures (cont.)

3. Conclusions of SSO Study

Although many SSO approaches were studied for this work, none fully satisfied the current needs for the MYSEA project. In particular, MYSEA operates in a MLS environment with both mandatory and discretionary access controls; these security characteristics were not found in the SSO solutions that were reviewed. Another security assumption for MYSEA is that clients are assumed to be untrusted, thus the requirement of the use of Trusted Path Extensions to perform security-critical transactions (e.g., authentication). To provide assurance that the TPEs are functioning securely and correctly, security analysis of TPEs is required. This analysis requires that the code for the TPE be understandable and is facilitated through minimization of the code base. Nearly all the above solutions require a significant amount of code for the clients. In addition, it is desired that the MYSEA servers be minimally impacted by the introduction of a SSO solution, but again, most of the SSO solutions studied required drastic changes to the server code.

For the reasons discussed above, a new single sign-on framework for the MYSEA environment will need to be developed. The remainder of this thesis is devoted to defining this framework for SSO in the MYSEA environment.

D. SUMMARY

This chapter presented a brief background on MYSEA and its components, an overview of single sign-on concepts and architectures, and a comparison of various SSO architectures. It concluded with the decision that a different kind of SSO architecture will need to be constructed to meet the needs of the MYSEA environment. The next chapter will describe the new SSO architecture and concept of operations, as well as provide an analysis of the requirements imposed by the new SSO solution.

III. MYSEA SSO REQUIREMENTS ANALYSIS

A. INTRODUCTION

The existing Monterey Security Architecture (MYSEA) does not support single sign-on (SSO) capabilities. This chapter describes a framework in which SSO can be incorporated into MYSEA. First, the system architecture and concept of operations (CONOPS) for SSO support is defined, including situations where some part of the SSO system has failed. This is followed by a description of an analysis of the threats to the system, based on the environmental assumptions, and a description of the organizational security policies. From these assumptions, threats, and policies, a set of objectives will be determined, and these objectives are used to derive the requirements for SSO support in MYSEA.

B. SYSTEM ARCHITECTURE

1. System-Supported Services

The MYSEA system architecture involves the management of the following elements: application services, authentication, trusted path extension, trusted channel, and service management. Application services are the user applications hosted by MYSEA servers, such as web, mail, and network file system. Authentication is the verification that a person is a legitimate user in the system and is used to control access to resources requested on MYSEA servers. Authentication in MYSEA also requires validating the session level of the user at which the user wishes to operate. Authentication may be distributed among multiple servers. The distribution of authentication to provide single sign-on is the main focus of this thesis.

Trusted path extension is the means in which a user located at a remote MYSEA client can establish a trusted path with the MYSEA server. This mechanism is implemented by a high assurance device called the Trusted Path Extension (TPE) that is attached to each MYSEA client and is an extension of the Trusted Computing Base (TCB) on the high assurance MYSEA server. Secure communications between any two components, (e.g., client to server, server to server) is achieved through a trusted channel. The trusted channel protects communications with respect to confidentiality, integrity, and authenticity. In the existing MYSEA design, the trusted channel is called the

Protected Communications Channel (PCC). Service management is the ability to configure the system services, such as the allocation of specific applications to particular servers and certain security within the overall system (e.g., the cryptographic algorithm used in communications). Service management is used to dynamically control the parameters of these services based on the operational conditions which can change over time (for example, component failures).

2. System Components

Presently, there is only a single MYSEA server that authenticates and services MYSEA clients. Additional MYSEA servers may be added in the future to support a larger number of clients and applications. Assuming no modifications to the current MYSEA authentication scheme for each application requested, the user operating a MYSEA client equipped with a Trusted Path Extension (TPE) must authenticate to each individual MYSEA server hosting the desired application. For example, if the user wishes to use three different applications, each hosted on a separate machine, the user is required to authenticate at least three times, once for each server. The person is forced repeatedly to enter authentication information, e.g., type her password, within a single session, potentially causing frustration. The root of this frustration is the fact that the MYSEA servers in the local area network (LAN) are presently designed to be autonomous, isolated machines with little if any communication amongst them.

The addition of single sign-on capabilities (SSO) in MYSEA forces these servers to be connected as a federation, rather than a loose grouping of separate entities. In this federation, MYSEA servers will be share authentication data for facilitation of SSO. One of the MYSEA servers in the federation will be assigned the role of Authentication Server (AUS), the central focal point of authentication. All MYSEA clients, through the TPEs, will authenticate directly to the AUS. A person using the MYSEA client equipped with a TPE need only authenticate once at a particular session level to the AUS in order to access several network applications that may be hosted on separate MYSEA servers. But the TPE will need to know what server to contact based on the application requested, so the TPE must now become protocol-aware. In contrast, the TPE in the current scheme handles all application requests by sending them to the same MYSEA server – the TPE is completely oblivious to the type of application that was requested.

Each of the other MYSEA servers, which host one or more network applications, will be known as an Application Management Server (AMS). The AUS can also be an AMS if it too hosts an application. An AMS provides services only to MYSEA users who have been authenticated. Previously, the AMS directly authenticated the users, requiring the user to enter the authentication information. But in this new federation, the AMS can query the AUS for user authentication information. The fact that a user has been authenticated at a particular session level has been captured by the AUS and this fact can be shared amongst the AMSes when they need to know if a particular user requesting their application has been authenticated.

The MYSEA clients with TPEs, the MYSEA authentication server, and the various MYSEA application servers are networked in a multilevel secure (MLS) local area network (LAN). The TPEs control each client's access to the LAN, and the TPE-AUS and TPE-AMS communications occur on this MLS network. The server-to-server communications, i.e., the AUS-AMS communications, also use the same MLS LAN. In the future, the server-to-server communications may occur on a separate LAN that is primarily used for the sharing of user authentication and session information, but the analysis and design of such a mechanism is out of the scope of this thesis.

In summary, the system architecture will now consist of MYSEA clients equipped with protocol-aware TPEs and a federation of MYSEA servers - a single Authentication Server (AUS) and a number of Application Management Servers (AMSes). The TPEs communicate with the AUS and AMSes through a MLS LAN. This LAN is also used for the AUS and AMS communications. Figure 1 shows the new MYSEA system architecture. The next section elaborates how the TPEs, AUS, and AMSes interact to accomplish single sign-on.

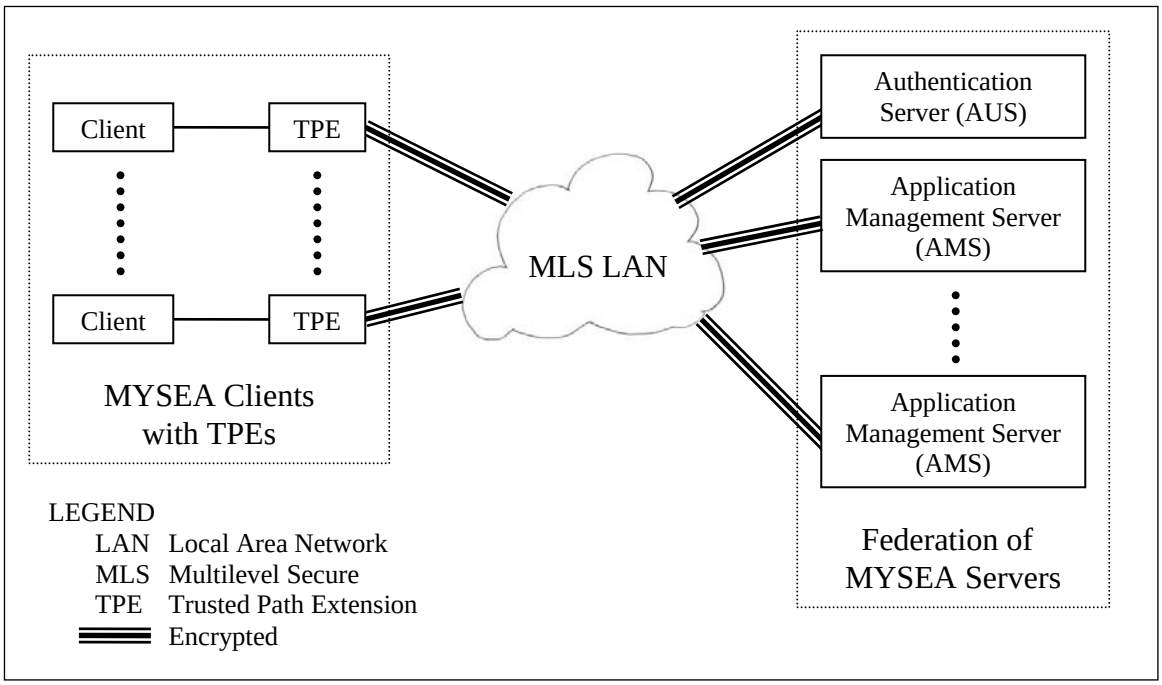

Figure 1. New MYSEA System Architecture

C. CONCEPT OF OPERATIONS

1. Initial User Authentication

Users must authenticate to the Authentication Server (AUS) prior to accessing the network and any network applications. A Trusted Path Extension (TPE), attached to each client machine, creates a secure, unforgeable communications path to the AUS that is used for user authentication and other security services. The user authentication process is depicted in Figure 2. For the sake of brevity, some steps in the authentication process have been consolidated.

First, the user presses the secure attention key (SAK) on the TPE. This causes the TPE to invoke a trusted path to the AUS and establish a secure connection with the AUS (Step 1). Next, the AUS issues to the user on the TPE a login prompt requesting the username and password (Step 2). The user then enters her username and password on the TPE, which sends these authentication items to the AUS (Step 3). The AUS checks the user's credentials; if they are valid, the AUS sends the TPE a message indicating the user has been authenticated at the default session level (Step 4).

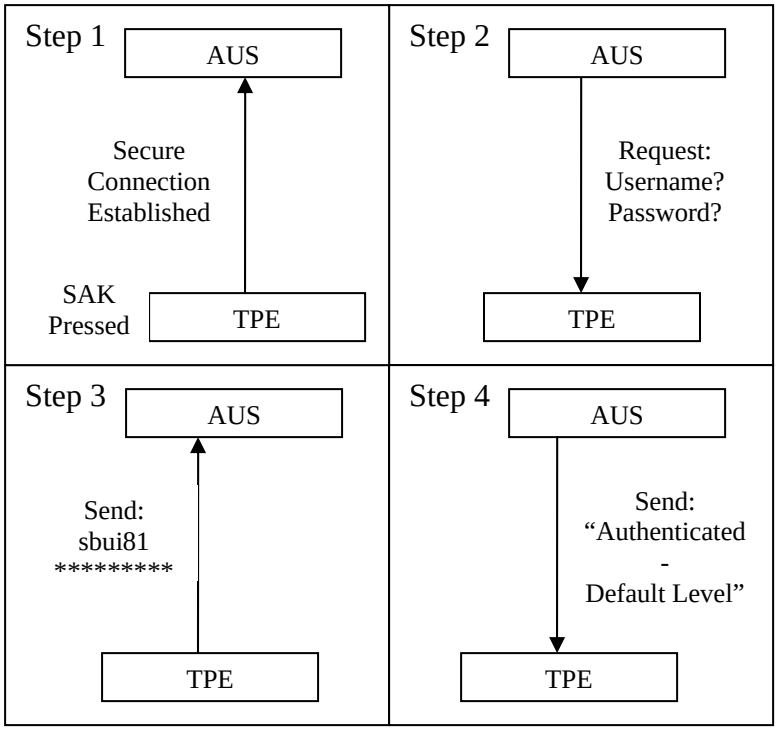

Figure 2. User Authentication Process

Next, the user presses the SAK on the TPE (Step 5 in Figure 3) and the AUS returns to the TPE a menu of options (Step 6), including an option to allow the user to start a session and an option to change the session level. If the user decides to operate at the default session level and wishes to use an application, the user selects the 'Run' option in (Step 7a) and in response, the AUS returns to the TPE an Application Mapping Database (Step 8a) that tells the TPE which server to contact for a particular network application (Step 9a). The existence of the Application Mapping Database (AMDB) is a new element in the MYSEA design, a part of the Service Management for MYSEA. More information about the AMDB will be given in the system requirements section of this chapter.

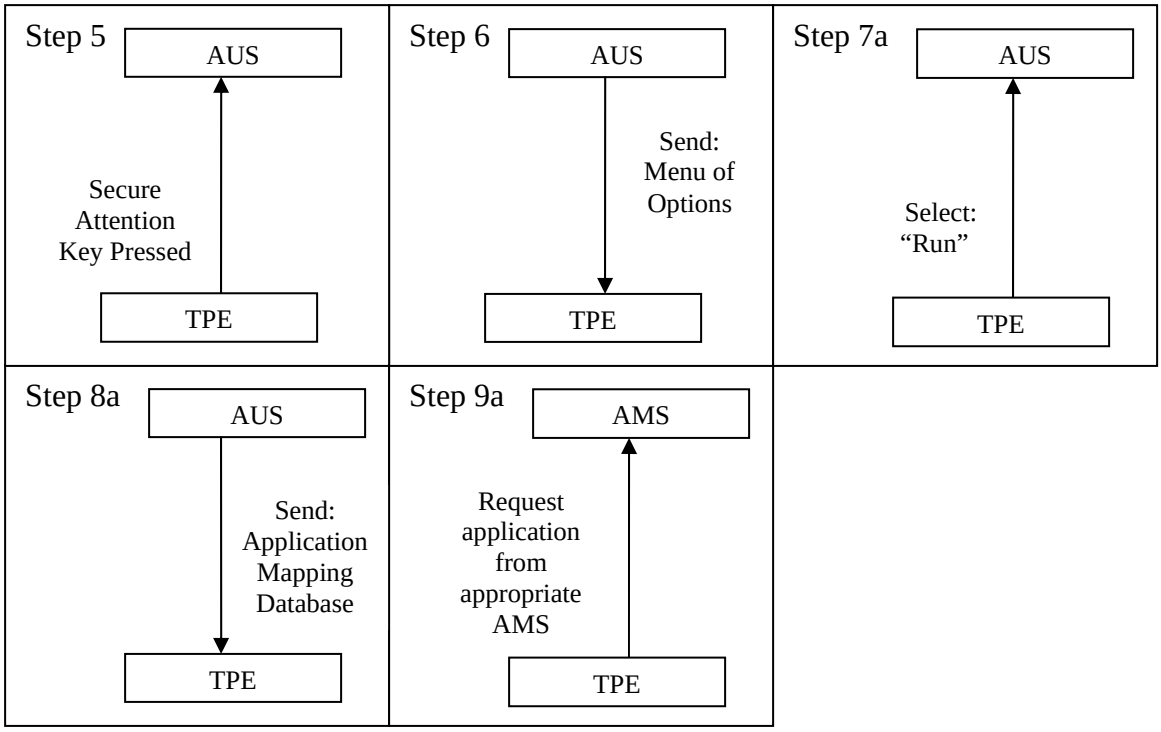

Figure 3. Selecting the Run Option

If the user wishes to change session levels, the user selects that option (Step 7b in Figure 4), and the AUS sends a prompt for the session level desired (Step 8b). The user enters the session level she wants to work at for the current session and the TPE sends this to the AUS (Step 9b). The AUS checks the user's clearance and if the user's clearance allows the requested level, returns a message to the TPE that the user is now operating at the new session level (Step 10b). If the user is ready to run an application, the user proceeds with the steps shown in Figure 3 starting at Step 7a.

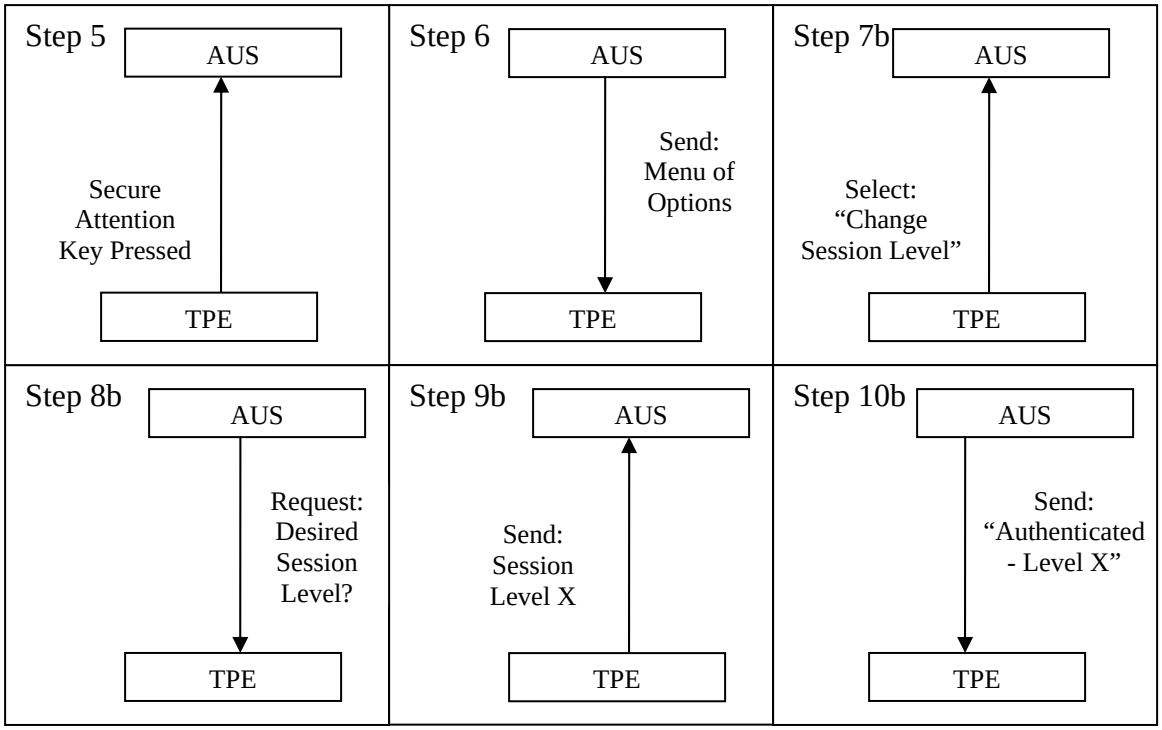

Figure 4. Changing the Session Level

At this point, the user on the MYSEA client is ready to run applications serviced by various MYSEA servers. The next section details the single sign-on process that allows the user to access applications on multiple servers.

2. Single Sign-On Process

After a user has successfully authenticated to the AUS, the user can access network applications running on the various Application Management Servers (AMS) that are part of the MYSEA MLS LAN. (The AUS could also be an AMS if it also hosts an application.) Figure 5 shows the process of the user accessing an application on the AMS. This is a single sign-on process because the authentication of the user to the AMS does not involve the user reentering her username and password.

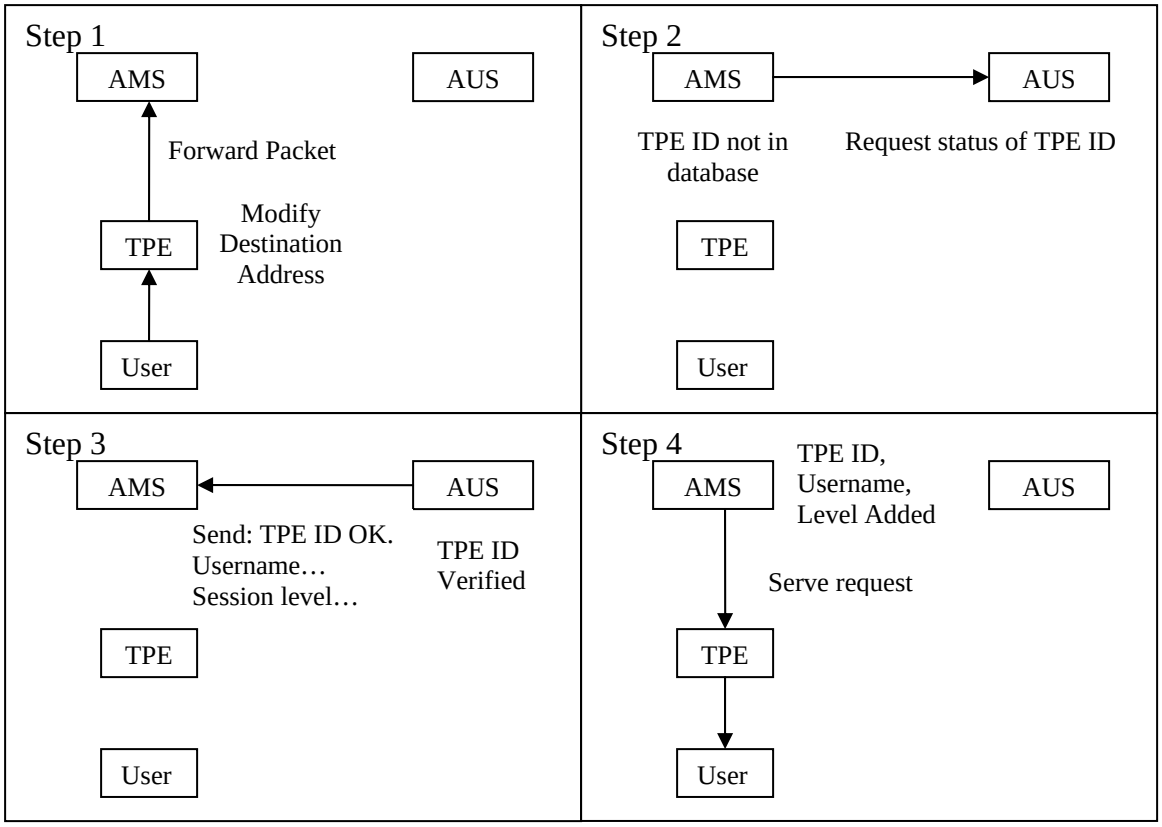

Figure 5. User Accessing an AMS Application

The user opens a network application. For example, the user opens the browser on the client machine to do web browsing. The application request causes the client machine to generate a network packet that specifies the type of the service requested (i.e., the destination port in the transport layer header). The TPE, which controls the client's access to the LAN, takes the packet, examines the service port, consults the Application Mapping Database for the address of the AMS hosting the requested service, and modifies the destination address of the packet to that of the appropriate AMS. The TPE then sends the packet to that AMS (Step 1). The ability of the TPE to examine the network packets at the transport layer level to determine the destination service port and redirect the packet to an AMS hosting the service is a feature that is not present in the current TPE design.

The AMS receives the network application request and checks if the ID of the TPE the user is working on is already present in its local authentication database. The authentication database on the AMS contains current session information of users who

have authenticated to the AUS and have accessed this AMS at least once during their current session. The current session information in the authentication database includes the user's current session level, the TPE ID associated with that user, and any other user attributes to support the discretionary access control (DAC) policy. Since this is the user's first application request to this AMS, the TPE ID of the user will not be present in the AMS's authentication database. In order to determine the login status of the user, the AMS must consult the AUS regarding the requesting user (Step 2) based on the user's TPE ID.

The AUS checks if the TPE ID is in its authentication database, which contains all authenticated users, their corresponding TPE IDs, and their current session level. The AUS then sends the AMS the results of the check – an affirmative that the TPE ID in question is present in the database, meaning that the user has been authenticated to the AUS using a TPE with that TPE ID, along with the username, the user's current session level, and other relevant user attributes (Step 3).

The AMS adds the user's TPE ID, username, her session level, and other pertinent user attributes to its local authentication database. The AMS then serves the user's application request (Step 4). On subsequent application requests within the same session, the AMS will only need to check its local authentication database to find the user's TPE ID, username, her current session information, and other user attributes that the AMS uses for allowing access to its services.

If the user wishes to use an application on another AMS, the above four steps are repeated. Single sign-on is achieved because the user does not have to repeatedly authenticate (i.e., enter a password) for each application server she has accessed.

3. User Session Status Update Process

Whenever the user wishes to change the status of her session, either by switching to a different session level or logging out and closing a session, the user presses the secure attention key on the TPE. Again, the TPE creates a trusted path to the AUS. The AUS sends to the TPE a menu of options which include logout, session level change, and resume the current session. The user enters the desired option into the TPE and the TPE sends the option to the AUS.

- If the option is to resume, the TPE is instructed by the AUS to continue with the current session. The AUS may also send to the TPE an updated Application Mapping Database (AMDB) to reflect any changes to the status of the AMSes.
- If the option is to logout, the AUS instructs the TPE to tear down all application connections associated with that user and to purge its memory. (A similar process must be done on the client machine to ensure that "all information is irretrievably removed from those objects/subjects" before its allocation for reuse [12]. However, the mechanisms to perform this are beyond the scope of this thesis.) The AUS then updates its authentication database to reflect that the user has logged out.
- If the option is to change the session level, the AUS sends the TPE a prompt for the user to enter the desired session level. The user enters the session level which the TPE sends to the AUS. The AUS checks if the session level is valid for the user - if it is a valid level, the AUS sends the command to the TPE to tear down any existing application connections, purge its memory, and run at the new session level. Again, the AUS may send a new AMDB to the TPE if the AMDB has changed.

In the case of logout or session level change, the AUS notifies the various AMSes accessed by the user in the last session that the user's login status has changed and to update their local authentication databases by flushing the user's old authentication information in their local databases. (This requires the AUS to keep track for each user of all the servers accessed in the current session. Whenever an AMS requests the user's authentication information from the AUS, the AUS adds the requesting AMS to the list of AMSes the user has accessed in the current session.) If the user changed session levels and wants to use an application on a particular AMS, the user will need to be authenticated to the servicing AMS through the single sign-on process as described in the previous section. In other words, the AMS will need to consult the AUS when the user makes the initial request because the user's current record will not be present in its local authentication database.

4. Failure Recovery

The MYSEA single sign-on system must be robust enough to handle a number of failure scenarios. These scenarios include TPE failure, Authentication Server failure, Application Management Server failure, Service Management failure, and network failure. Combinations of failure scenarios are also possible but the analysis of such combinations is left as future work.

TPE failure is when the TPE is in an unknown, unstable state and either is unresponsive or powered off. In this scenario, the TPE is no longer communicating with the AUS or any AMS during a user's session (the user has not logged out from a particular client with a TPE). After an administratively defined TPE timeout interval, the AUS will query each of the AMSes accessed by the user for recent activity. Each of those AMSes will check the activity timeouts (administratively defined) each has set for the TPE in question. If the timeout period on an AMS has not occurred, the AMS replies to the AUS that the TPE is active, and the AUS will reset the TPE timeout interval to the maximum defined by the administrator since the TPE is still functioning normally. If all the AMSes reply that the user has not been active, an attempt to contact the TPE at the network level will be made by the AUS. In the case that the TPE cannot be contacted through the TPE heartbeat mechanism, the AUS will remove the failed TPE and the logged in user from the authentication database. The AUS also will alert the AMSes that have served the user in the current session to remove the user from their local authentication databases. Future work includes designing this TPE heartbeat mechanism to detect TPE failure at the network level.

AMSes may also be able to detect TPE failures and inform the AUS of the failure, but this scenario is left as future work. For the current work, the AUS is responsible for detecting TPE failure and informing AMSes of the failure.

It is also possible that a failure may occur at the Authentication Server. The failure of the AUS will be detected by an AMS requesting a user's authentication information because the AUS will not respond to the AMS's request. The AMS will determine that the AUS has failed after an administratively set number of non-responses by the AUS. The AMS may also detect that the AUS is down during heartbeat

monitoring. Heartbeat monitoring is used to determine if a party is alive, in this case the AUS. The details of the heartbeat monitoring mechanisms is out of the scope of this thesis but will be addressed in future work.

In the case of AUS failure, no new users will be able to authenticate to the AUS and access the LAN. An authenticated user will not be able to access an AMS if the AMS did not contact the AUS for her authentication status before the AUS failed, nor can an authenticated user change her session level during an AUS failure. Active users, authenticated users who are actively using one or more applications, will be able to continue using those applications, but will not be able to change session levels or connect to new applications running on servers that were not accessed by the user before the failure in the AUS. Idle users still connected to an AMS will be disconnected by the AMS after an administratively set timeout interval. Increasing the availability of the AUS and AMS services may include appointing an AMS to serve as an alternate AUS, but the design of this approach is not in the scope of this work.

An Application Management Server failure can happen if any part of the AMS fails, such as the server application, server operating system, hardware, etc. AMS failures can be detected through heartbeat monitoring mechanisms between the AMSes and the AUS. The AUS will update its Application Mapping Database (AMDB) and send the TPEs that were using the failed AMS the updated AMDB. In addition, a secure attention key press on the TPE will also trigger receipt of the updated AMDB.

Service Management failure can occur if the Application Mapping Database sent to a TPE is not valid, i.e., the TPE is directed to an AMS that is not hosting the application as specified in the database. To deal with this inconsistency, the TPE must contact the AUS, through the user's pressing of the SAK to obtain the latest AMDB before starting the session with an AMS. However it is possible that the AMDB may become inconsistent after the TPE has downloaded the database. The handling of this type of failure is outside the scope of this thesis and will be discussed in the future work section.

The concept of operations (CONOPS) for the single sign-on system in MYSEA was described. The CONOPS included normal-use as well as failure scenarios. The next section defines the single sign-on system.

D. SINGLE SIGN-ON SYSTEM DESCRIPTION

The Common Criteria version 2.2 [12] was used as the basis for generating the requirements for the single sign-on system. The Common Criteria provides a standard methodology for the specification of security requirements for IT products and their evaluation at specific security assurance levels. For more information on the Common Criteria, refer to the website listed in [13].

The Common Criteria term for the IT product that is to be defined and evaluated is Target of Evaluation (TOE), and the part of the TOE that performs security functions of the TOE, such as identification and authentication, is called the TOE Security Functions (TSF). Before any security requirements can be generated, the TOE and TSF must be clearly defined. The TOE, within the MYSEA single sign-on context, is the Authentication Server (AUS). The AUS is a specialized MYSEA server so it runs on the same hardware platform and operating system as any MYSEA server (DigitalNet XTS-400 running the STOP operating system). This TOE includes the following MYSEA services: Trusted Path Service (TPS), Secure Session Service (SSS) if it also acts as an AMS, Dynamic Security Service (DSS), and Trusted Channel Service (TCS), as well as any administrative tools. The TPS, SSS, DSS, TCS were described in the Chapter II. Other MYSEA servers also contain these MYSEA services, so the AUS's single sign-on capabilities distinguishes it from these other MYSEA servers. The TOE Security Functions (TSF) for this work includes the hardware, operating system, SSS, and TPS components of the AUS.

The AUS performs all remote user authentication and session handling. The information about authenticated users is stored in databases on the AUS, and this information is shared by the AUS with the various Application Management Servers (AMS). The AUS is also responsible for performing service management activities such as giving each TPE (attached to a MYSEA client) the Application Mapping Database that gives the location of the AMS hosting the service and verifying that the AMSes are active.

For this work, the AUS is a single MYSEA server. There may be multiple AUSes for increased scalability and reliability, but the analysis and design of a SSO system with multiple AUSes, such as the load balancing and consistency mechanisms, is left as future work. However, this current design, with one AUS, must include requirements that will allow the extension of the design to multiple AUSes.

The TOE, or AUS, is also a component in a larger system – the MYSEA MLS LAN. In this context, the AUS supports system-wide services, such as SSO and heartbeat monitoring. This work focuses on specifying requirements for the AUS to support single sign-on. Preliminary requirements for heartbeat monitoring will also be specified, but a more detailed specification for heartbeat monitoring is out of the scope of this work. The guidance on how to specify the TOE as a stand-alone system as well as a component in a larger system was based on the U.S. Government Directory Protection Profile for Medium Robustness Environments [14] which specified requirements for a directory server (such as that used in a public key infrastructure) that is both a separate system and a component in a larger system.

The remainder of this chapter will detail the assumptions, threat analysis, organizational security policies, objectives and system level requirements for the single sign-on system. The Consistency Instruction Manual for Medium Robustness Environments [15] and the Directory Protection Profile [14] were referenced in developing these components. It is standard practice to directly use sections from the CIM and other relevant Protection Profiles when applicable to the current system.

E. ASSUMPTIONS

The table below lists the assumptions of the Authentication Server, or TOE, security environment. Assumptions are aspects of the environment that the TOE is not able to control, such as physical security and the trustworthiness of the remote components (e.g., TPEs and AMSes) that interact with the TOE.

Assumption Name	Assumption Description
A.PHYSICAL	It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.
A.REMOTE_COMPONENT_ENVIRONMENT	The accreditation process will ensure that the procuring organization will manage and protect the remote components in a manner that is commensurate with the protection mechanisms provided by the TOE.
A.REMOTE_COMPONENT_FUNCTIONALITY	Remote component processes that interact with the TOE are trusted to comply with the security requirements levied upon them by the TOE.
A.NETWORK_SECURITY_POLICY_ENFORCEMENT	All network components are vetted with the appropriate level of trust in order to properly enforce the network security policy.

Table 3. System Assumptions

F. THREAT ANALYSIS

The threat analysis involves identifying the relevant threats to the TOE. A table of threats was compiled using the Consistency Instruction Manual for Medium Robustness Environments [15]. These threats were examined for their relevance to the Authentication Server and those that were appropriate to this TOE are described to address specific aspects of Authentication Server.

Threat Name	Threat Description
T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.
T.ADMIN_ROGUE	An administrator's intentions may become malicious resulting in user or TSF data being compromised.
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.CONFIG_CORRUPT	A malicious process, user, or external IT entity may cause configuration data or other TSF data to be lost or modified.
T.CRYPTO_COMPROMISE	A malicious user or process may cause keys, data or executable code associated with the cryptographic

	functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
T.EAVESDROP	A malicious user or process may observe or modify user or TSF data transmitted between the TOE and a remote entity.
T.MASQUERADE	A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources.
T.POOR_DESIGN	Unintentional or intentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a malicious user or program.
T.POOR_IMPLEMENTATION	Unintentional or intentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a malicious user or program.
T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.
T.REPLAY	A user may gain inappropriate access to the TOE by replaying authentication information, or may cause the TOE to be inappropriately configured by replaying TSF data or security attributes (e.g., captured as transmitted during the course of legitimate use).
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.RESOURCE_EXHAUSTION	A malicious process or user may block others from system or network resources (e.g., network applications) via a resource exhaustion denial of service attack.
T.SPOOFING	A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.TSF_ COMPROMISE	A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ACCESS	A user may gain access to user data for which they are not authorized according to the TOE security policy.

T.UNIDENTIFIED_ACTIONS	The administrator may fail to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.
T.UNKNOWN_STATE	When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.

Table 4. System Threats

G. ORGANIZATIONAL SECURITY POLICIES

The policies of the organization define how the system is to be used, which includes the limits as well as capabilities. The table below shows the policies needed to establish single sign-on capabilities in a medium-robustness environment.

Policy Name	**Policy Description**
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.LOCAL_ADMIN_ACCESS	Administrators shall be able to administer the TOE locally only.
P.CRYPTOGRAPHY	The TOE shall use NIST FIPS validated cryptography as a baseline with additional NSA-approved methods for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys), and for cryptographic operations (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services).
P.SINGLE_SIGN_ON	Authorized users shall be able to access services on a federation of servers after successful authentication.
P.VULNERABILITY_ ANALYSIS_TEST	The TOE shall undergo appropriate independent vulnerability analysis and penetration testing to demonstrate that the TOE is resistant to an attacker possessing a medium attack potential.

Table 5. Organizational Security Policies

H. OBJECTIVES

1. Security Objectives for the System

The following table lists the security objectives for the system in order to address the threats and policies. These objectives will lead to the generation of security requirements to achieve these objectives.

Objective Name	Objective Description
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security-relevant events associated with users.
O.ADMIN_ROLE	The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information by controlling access to the audit trail.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.
O.CHANGE_MANAGEMENT	The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.
O.CORRECT_ TSF_OPERATION	The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.
O.CRYPTOGRAPHY	The TOE will use NIST FIPS 140-2 validated cryptographic services.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.MAINT_MODE	The TOE will provide a mode from which recovery or initial startup procedures can be performed.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.
O.PROTECT_IN_TRANSIT	The TSF will protect user and TSF data when it is in transit from the TOE to another

	remote entity.
O.REPLAY_DETECTION	The TOE will provide a means to detect and reject the replay of authentication data as well as other TSF data and security attributes.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.
O.RESOURCE_SHARING	The TOE will provide mechanisms that mitigate attempts to exhaust CPU resources provided by the TOE (e.g., network authentication.).
O. ROBUST_ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure delivery and management of the TOE.
O. ROBUST_TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.
O.SELF_PROTECTION	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure.
O.SINGLE_SIGN_ON	The TOE will provide a means to ensure users will be able to access services on a federation of servers after successful authentication.
O.SINGLE_SIGN_ON_SUPPORT	The TOE will provide either centralized or distributed user identification and authentication mechanisms that are secure.
O.SOUND_DESIGN	The TOE will be designed using sound design principles and techniques. The TOE design, design principles, and design techniques will be adequately and accurately documented.
O.SOUND_IMPLEMENTATION	The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.
O.THOROUGH_FUNCTIONAL_ TESTING	The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.
O.TIME_STAMPS	The TOE will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.
O.TRUSTED_CHANNEL	The TOE will provide a means to establish protected communications with remote entities based on the security attributes of the

	remote entity.
O.TRUSTED_PATH	The TOE will provide a means to ensure that users are not communicating with some other entity pretending to be the TOE when supplying identification and authentication data.
O.USER_GUIDANCE	The TOE will provide users with the information necessary to correctly use the security mechanisms.
O.VULNERABILITY_ANALYSIS_TEST	The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.

Table 6. System Security Objectives

2. Security Objectives for the Environment

The security objectives of the environment state the goals that must be addressed by the IT environment. Table 7 is a summary of the objectives that are imposed on the SSO environment.

Environmental Objective Name	Environmental Objective Description
OE.NETWORK_SECURITY_POLICY_ENFORCEMENT	The security administrator must ensure that the appropriate level of trust has been established among all components such that network security policies are understood and enforced.
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.
OE.PROTECTED_COMMUNICATIONS_CHANNEL	Remote authorized IT entities must provide a means to communicate securely with the TOE.
OE.REMOTE_COMPONENT_ENVIRONMENT	The accreditation process will ensure that the procuring organization will manage and protect the remote components (i.e., Application Management Servers, Trusted Path Extensions) in a manner that is commensurate with the protection provided for the TOE.
OE.REMOTE_COMPONENT_FUNCTIONALITY	Remote component trusted processes will be constructed to comply

	with the security requirements levied upon them by the TOE.

Table 7. Operational Environment Security Objectives

I. SYSTEM LEVEL REQUIREMENTS

This section discusses an initial set of requirements for single sign-on in the MYSEA environment. The system level requirements consist of single sign-on requirements and service management requirements. These requirements are further divided among separate requirements for the Authentication Server, Trusted Path Extensions (TPE), Application Management Servers, administrators and the users. Any requirement that depends on communications between two entities (e.g., TPE and AUS, AUS and AMS, etc.) implies that the entities communicate through the Protected Communications Channel (PCC).

1. Single Sign-on Requirements

a. Authentication Server Requirements

- Before user authentication, the AUS shall authenticate TPEs based on TPE ID stored in the Allowed TPE Database.
- The AUS shall authenticate users communicating through valid TPEs based on a user ID and password stored in the password file (managed by the operating system).
- Upon the SAK press, the AUS shall provide to valid TPEs with authenticated users a menu of available functions that the users can invoke. The list of functions includes: change session level, start a session with an AMS, and logout.
- The AUS shall maintain the AMDB. This database shall contain information about available services provided by different AMSes.
- The AUS shall construct a subset of the Application Mapping Database (AMDB) for each TPE. The AUS shall distribute the AMDB subset to the TPE upon successful user authentication. The AUS shall send AMDB updates to the TPE when the SAK is pressed during a user's session.

- The AUS shall maintain the User Database to keep track of the status of currently authenticated users. User status information includes the user ID, TPE ID, current session level, and any other user attributes used for access control decisions.
- The AUS shall be able to receive user session information requests from AMSes.
- Before responding with a user's session information from the User Database, the AUS shall authenticate AMSes based on the AMS ID.
- For each user, the AUS shall maintain a list of AMSes that have requested the user's session information. The AUS shall use this list to inform the appropriate AMSes of user logout or session level change and to inform TPEs of changes in their AMDB subsets (i.e., AMS no longer servicing an application).
- The AUS shall have configurable auditing capabilities. The events that may be audited include: user authentication, AMS user authentication request, and distribution of the AMDB subset.

b. Trusted Path Extension Requirements

- The TPE shall provide an interface for the user to authenticate and negotiate a session level with the AUS.
- The TPE shall communicate the information provided by the user to the AUS.
- The TPE shall provide access to the LAN only after the user has authenticated to the AUS.
- The TPE shall be able to receive a subset of the Application Mapping Database (AMDB) from the AUS upon successful user authentication.

- The TPE shall send an acknowledgement of the receipt of the AMDB subset to the AUS.
- The TPE shall route user application requests to the appropriate AMS based on the AMDB received from the AUS.

c. Application Management Server Requirements

- The AMS shall validate the user before servicing the user's application request.
- The AMS shall make the decision to service the requests based on the user session information (e.g., user ID, TPE ID, current session level) in its cached User Database.
- If the user's information is not in the AMS's User Database cache, the AMS shall contact the AUS to obtain the most current authentication information of the user and store it in the AMS's User Database cache.
- The AMS shall be able to receive user session updates from the AUS and update the AMS's User Database cache as appropriate.
- The AMS shall notify the AUS when a service becomes unavailable.

d. Administrator Requirements

- The administrator shall configure the databases on the AUS such that the AUS is able to authenticate TPEs, users, and AMSes.
- The administrator shall configure the TPEs and AMSes such that they will be able to communicate with the AUS.

e. User Requirements

- All users shall be registered with the administrator before they are given access to the LAN.

2. Service Management Requirements

a. Authentication Server Requirements

- The AUS shall maintain consistency of the Application Mapping Database. The AUS shall monitor the AMSes for liveliness and the types of applications actively hosted by the AMS.
- The AUS shall detect AMS failure and notify the affected TPEs in a timely manner by sending them updated AMDB subsets.
- The AUS shall maintain the consistency of the subsets of the AMDB on the TPEs.
- The AUS shall query the appropriate AMSes when a TPE has not communicated with the AUS within the set timeout interval.
- The AUS shall detect TPE failure and contact the appropriate AMSes to remove the user associated with the TPE from their local User Database caches.

b. Trusted Path Extension Requirements

- The TPE shall update its copy of the AMDB subset when it is received from the AUS.
- The TPE shall respond to AUS monitoring mechanisms.

c. Application Management Server Requirements

- The AMS shall remove a user from its User Database cache after an administratively set timeout interval for non-activity (i.e., the user has not used the AMS within the timeout interval). The timeout interval shall be reset to the maximum value whenever the AMS services the user.
- The AMS shall respond to the AUS queries of recent TPE activity on the AMS within a timeout interval.
- The AMS shall respond to AUS requests for AMS status.

- The AMS shall detect AUS failure through monitoring mechanisms.
- In the event of AUS failure, for each user still present in its local User Database cache, the AMS shall continue servicing until the user has closed the application session or has timed out due to non-activity.

d. Administrator Requirements

- The administrator shall configure the monitoring mechanisms on the AUS and AMSes.
- The administrator shall configure the AMS's user non-activity timeout interval.
- The administrator shall configure the AUS's timeout interval for TPE non-activity.
- The administrator shall configure the TPE's user non-activity timeout.

J. SUMMARY

This chapter described the various components, issues, and mechanisms for incorporating single sign-on in MYSEA. The system architecture, concept of operations, assumptions, threats, policies, objectives, and system requirements were discussed. The next chapter focuses on enumerating the security requirements for the Authentication Server, the core of the single sign-on system for MYSEA.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SECURITY REQUIREMENTS

A. INTRODUCTION

This chapter consists of the initial set of security requirements for the Authentication Server in the MYSEA single sign-on system. These requirements are based on the threats, assumptions, policies, and objectives from the previous chapter. The methodology for generating requirements used in this project is based on the Common Criteria (CC) v2.2 [13], the Consistency Instruction Manual (CIM) for Medium Robustness Environments [15], and a CC-based requirements derivation framework for informally defined systems [16]. It is standard practice to list applicable requirements verbatim from these sources to ensure completeness and consistency of the requirements. Some requirements have been modified and augmented to reflect the requirements for this work.

The requirements are divided into two categories: security functional requirements and security requirements. Following the requirements is a mapping of security threats, policies, and assumptions to the objectives. The objectives are then mapped to the security requirements.

B. AUTHENTICATION SERVER SECURITY FUNCTIONAL REQUIREMENTS

1. Authentication Server Audit

1.1 The Authentication Server (AUS) shall have configurable auditing capabilities. The levels of auditing are hierarchical, from the least amount of audit information to the most. The Authentication Server will support the following audit levels (from high to low): alert, critical, error, warning, notice, information, and debugging. All audited events shall be recorded.

1.2 The types of events that shall be audited include user authentication, Application Management Server (AMS) retrieval of user authentication information, transfer of an Application Mapping Database to a remote device, user session level change/logout, failure in a remote component, and reading/modification of the audit trail.

1.3 The date and time of the event, IP address of the remote host, type of event, user name (if applicable), user session level (if applicable), and the event outcome (success or failure) shall be recorded.

1.4 The AUS shall alarm the Security Administrator whenever a security violation has been violated through a message displayed on the local console identifying the violation and allow access to the audit records associated with the event.

1.5 The AUS shall display an acknowledgement message identifying a reference to the potential security violation, a notice that it has been acknowledged, the time of the acknowledgement and the user identifier that acknowledged the alarm, at the local console.

1.6 Authorized administrators shall be the only entities able to review, delete, or modify the audit logs. Authorized administrators shall be able to configure the actions to take when an audit log is full.

1.7. The audit records generated by the AUS shall be in a format that can be parsed.

2. Authentication Server Communication

2.1 The AUS shall employ cryptographic functionality for secure connection between the AUS and remote IT entities (i.e., TPEs and AMSes).

3. Authentication Server Cryptography

3.1 The AUS shall use NIST-validated cryptographic standards when using cryptography for communications to remote IT entities.

3.2 The cryptographic keys used by the AUS shall be managed using NIST-validated mechanisms. This includes the generation and destruction of cryptographic keys.

3.3 The AUS shall use NIST-validated cryptomodules in a NIST-validated mode for cryptographic operations.

4. Authentication Server Data Protection

4.1 For all user accesses to AUS resources, the AUS shall enforce the access control policy and information flow policy based on the user ID and session level.

4.2 For all remote device accesses to AUS resources, the AUS shall enforce the access control policy and information flow policy based on the remote device ID (i.e., TPE ID, AMS ID) and security level.

4.3 The AUS shall ensure that any previous information content of a resource is made unavailable upon the resource's reallocation to any AUS objects.

5. Authentication Server Identification and Authentication

5.1 The AUS shall ensure that users are identified and authenticated in order to associate them with the proper security attributes, such as user name and session level, prior to access to AUS data or network applications.

5.2 The AUS shall authenticate registered users based on the user ID and a password.

5.3 The AUS shall ensure that Application Management Servers are identified and authenticated prior to access to AUS data.

5.4 The AUS shall authenticate registered Application Management Servers based on a digital certificate.

5.5 The AUS shall associate the following user security attributes with subjects acting on the behalf of that user: username, session level, and any other appropriate security attributes.

5.6 The AUS shall associate the following security attributes with subjects acting on the behalf of the remote component: hostname, host IP address, and any other appropriate security attributes.

5.7 The AUS shall detect when an administrator-configurable number of unsuccessful authentication attempts occur within an administrator-configurable time period.

6. Authentication Server Protection

6.1 The Authentication Server shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

6.2 The Authentication Server shall enforce separation between the security domains of subjects in the AUS scope of control.

6.3 The AUS shall ensure the availability of user session information provided to a remote component within a security administrator-configurable time given the remote component has been authenticated to the AUS and is functioning normally.

6.4 The Authentication Server shall detect replay of authentication information, other AUS data, or AUS security attributes transmitted during the course of legitimate use.

6.5 The AUS shall be able to provide reliable time stamps for its own use.

6.6 When automated recovery from failures/service discontinuities is not possible, the AUS shall enter a maintenance mode where the ability to return to a secure state is provided.

6.7 The AUS shall run a suite of self-tests during initial start-up, periodically during normal operation as specified by an authorized administrator, and at the request of an authorized administrator to demonstrate the correct operation of the software portions of the AUS.

6.8 The AUS shall ensure that security policy enforcement functions are invoked and succeed before each function within the AUS scope of control is allowed to proceed.

7. Authentication Server Resource Management

7.1 The AUS shall enforce administrator-specified maximum quotas of the AUS services (i.e., authentication, access to AUS databases) that users and remote entities can use over an administrator-specified period of time.

8. Authentication Server Security Management

8.1 The AUS shall restrict configuration of security services management, such as setting quota limits and audit configuration, only to authorized administrators.

8.2 The AUS shall maintain the roles: Security Administrator; Cryptographic Administrator (i.e., users authorized to perform cryptographic initialization and management functions); and Audit Administrator.

8.3 The AUS shall be able to associate users with roles.

9. Authentication Server Access

9.1 Before establishing a user session that requires authentication, the AUS shall display only an authorized administrator-specified advisory notice and consent warning message regarding unauthorized use of the AUS.

9.2 The AUS shall be able to deny session establishment based on the TPE ID, user ID, and user clearance.

9.3 The AUS shall provide AUS-initiated session locking after an administratively-set timeout interval.

10. Authentication Server Trusted Path/Channels

10.1 The AUS shall provide a communication channel between itself and remote components that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

10.2 The AUS shall permit itself or an authorized remote component to initiate communication via the trusted channel for remote user/component authentication, SSO service management, and other network security management functions.

10.3 The AUS shall provide a communication path between itself and users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

10.4 The AUS shall permit local users to initiate communication via the trusted path for initial user authentication and session negotiation.

11. Authentication Server Single Sign-on Management

11.1 The AUS shall maintain a user database containing information about currently authenticated users (e.g., user ID, session level, TPE ID, etc).

11.2 The AUS shall distribute information in the user database to authorized, authenticated remote entities (i.e., AMSes) when user authentication information is requested by the remote entity and when a user's session status changes (e.g., user logout or session level change).

11.3 The AUS shall maintain an Application Mapping Database that maps applications to hosting Application Management Servers.

11.4 The AUS shall distribute portions of the Application Mapping Database to authorized remote entities (i.e., TPEs) after successful user authentication, successful user session negotiation, and upon SAK press during a user's session when the AMDB subset on the TPE is inconsistent with the AUS's AMDB.

11.5. The AUS shall be able to provide single sign-on services whether it acts alone or with multiple AUSes.

11.6. The AUS shall be able to securely distribute single sign-on management information (e.g., user database, Application Mapping Database) to other authorized AUSes.

C. AUTHENTICATION SERVER SECURITY ASSURANCE REQUIREMENTS

The security assurance requirements for the existing MYSEA Server were developed according to current MYSEA practices. The following assurance requirements for the Authentication Server are based on those found in [16], which specified the design of an informally defined system, the Common Criteria assurance requirements, and those found in CIM.

1. Authentication Server Configuration Management

1.1 The software and documentation (e.g., design specifications, guidance documents) for the Authentication Server, including configuration files, shall be placed under configuration management (CM).

1.2 The CM system will uniquely identify configuration items, including those associated with the AUS (i.e., Authentication Server implementation and documentation).

1.3 The CM documentation shall include a configuration list, a CM plan, and an acceptance plan. The CM documentation shall provide evidence that the CM is maintaining the configuration items.

1.4. The acceptance plan shall describe the acceptance procedures for modified or newly created configuration items of the AUS.

2. Authentication Server Operation

2.1 The installation, generation and start-up documentation shall be provided and shall describe all the steps necessary for secure installation, generation and start-up of the AUS.

3. Authentication Server Development

3.1 An informal functional specification describing the interfaces of the security functions for the Authentication Server shall be provided.

3.2 An informal high-level design of the AUS shall be developed.

3.3 An informal architectural design of the AUS security functions shall be developed in detail sufficient to determine that the security enforcing mechanisms cannot be bypassed.

3.4 An informal low level design shall be developed for the AUS.

3.5 The design of the AUS shall meet the functional requirements.

3.6. The implementation of the AUS shall be an accurate instantiation of the design.

3.7 The implementation of the AUS shall be adequately and accurately documented such that the AUS can be generated without further design decisions.

3.8 An informal security policy model of the AUS shall be developed.

4 Authentication Server Guidance Documents

4.1 The user guidance shall describe the interaction between the user and the Authentication Server for proper authentication, session level modification, and logout.

4.2 The user guidance shall clearly present all user responsibilities necessary for secure use of the Authentication Server.

4.3 The administrative guidance shall describe the procedures and technical measures necessary to restrict physical access to the system.

4.4 The administrative guidance shall cover configuration, maintenance, and administration of the Authentication Server in a secure manner. The guidance is intended to help administrators understand the security functions of the Authentication Server,

including both those functions that require the administrator to perform security-critical actions and those functions that provide security-critical information to the administrator

4.5 The administrative guidance shall describe the functions and interfaces available to the administrator in addition to how to manage the AUS in a secure manner.

4.6 The administrative guidance shall describe all security requirements for the operational environment that are relevant to the administrator and the AUS.

5 Authentication Server Life Cycle Support

5.1 The AUS shall follow the same life cycle model and developmental procedures as the MYSEA project.

6 Authentication Server Test Coverage

6.1 The AUS and its security functions shall be tested to ensure that it operates in accordance with its high-level design and low-level design.

6.2 Test documentation (e.g., test plan, procedures, and results) shall be produced.

7 Authentication Server Vulnerability Assessment

7.1 Guidance documentation for the Authentication Server shall be complete, clear, consistent, and reasonable. The guidance documentation shall: identify all possible modes of operation of the AUS (including operation following failure or operational error), their consequences and implications for maintaining secure operation; list all assumptions about the intended environment; list all requirements for external security measures (including external procedural, physical and personnel controls); and demonstrate that the guidance documentation is complete.

7.2 The developer shall perform a vulnerability analysis and provide vulnerability analysis documentation.

D. THREAT AND POLICY MAPPING

The following table shows how the objectives are mapped to the threats, and a rationale for the mapping is given. This is followed by table for the mapping of objectives to the policies.

The terms TOE and TSF refer to the Authentication Server. These two terms are appear in the table below because the CIM was used as a basis for creating the mapping rationale, and it is customary to directly incorporate CIM material.

Threat Name	Objectives Addressing Threat	Rationale
T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management of the TOE.	O.ROBUST_ADMIN_GUIDANCE helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is insecure.
	O.ADMIN_ROLE The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally.	O.ADMIN_ROLE plays a role in mitigating this threat by limiting the functions an administrator can perform in a given role.
	O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE also contributes to mitigating this threat by providing administrators the capability to view configuration settings. For example, if the Security Administrator made a mistake when configuring the rule-set, providing them the capability to view the rules affords them the ability to review the rules and discover any mistakes that might have been made.
T.ADMIN_ROGUE An administrator's intentions may become malicious	O.ADMIN_ROLE The TOE will provide administrator roles to isolate	O.ADMIN_ROLE mitigates this threat by restricting the functions available to an administrator. This is somewhat different than the part this objective plays in countering T.ADMIN_ERROR,

resulting in user or TSF data being compromised.	administrative actions, and to make the administrative functions available locally.	in that this presumes that separate individuals will be assigned separate roles. For example, the Audit Administrator may detect malicious actions from Security Administrator.
T.AUDIT_COMPROMISE A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	O.AUDIT_PROTECTION The TOE will provide the capability to protect audit information by controlling access to the audit trail.	O.AUDIT_PROTECTION contributes to mitigating this threat by controlling access to the audit trail. The auditor and any trusted IT entities performing IDS-like functions are the only ones allowed to read the audit trail. No one is allowed to modify audit records, and the Auditor is the only one allowed to delete audit records in the audit trail. The TOE has the capability to prevent auditable actions from occurring if the audit trail is full, and of notifying an administrator if the audit trail is approaching its capacity. In addition, the TOE has the capability to restore audit data corrupted by the attacker.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION prevents a user not authorized to read the audit trail from access to audit information that might otherwise be persistent in a TOE resource (e.g., memory). By ensuring the TOE prevents residual information in a resource, audit information will not become available to any user or process except those explicitly authorized for that data.
	O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure.	O.SELF_PROTECTION contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the audit trail. Likewise, ensuring that the functions that protect the audit trail are always invoked is also critical to the mitigation of this threat.
T.CONFIG_CORRUPT	O.MAINT_MODE	O.MAINT_MODE mitigates this threat by providing a mode to recover from

A malicious process, user, or external IT entity may cause configuration data or other TSF data to be lost or modified.	The TOE will provide a mode from which recovery or initial startup procedures can be performed.	malicious modifications/deletions of TSF data or configuration data.
	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management of the TOE.	O.ROBUST_ADMIN_GUIDANCE is necessary to mitigate this threat by providing administrators the information for managing the TOE configuration data to protect against malicious actions by other entities.
	O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE addresses this threat by restricting the abilities to alter the configuration data or other TSF data to authorized administrators.
T.CRYPTO_COMPROMISE A malicious user or process may cause keys, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION is necessary to mitigate this threat by ensuring no TSF data remain in resources allocated to a user. Even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data.
	O.SELF_PROTECTION The TSF will maintain a domain	O.SELF_PROTECTION contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be

cryptographic mechanisms and the data protected by those mechanisms.	for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.	trusted to control access to the resources under its control, which includes the cryptographic data and executable code.
T.EAVESDROP A malicious user or process may observe or modify user or TSF data transmitted between the TOE and a remote entity.	O.PROTECT_IN_TRANSIT The TSF will protect user and TSF data when it is in transit from the TOE to another remote entity.	O.PROTECT_IN_TRANSIT ensures that both user and TSF data are protected in transit for modification and disclosure. This is achieved by using cryptography.
T.MASQUERADE A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources.	O. ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	O. ROBUST_TOE_ACCESS mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanisms, this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE. This objective also allows the TOE to correctly interpret information used during the authentication process so that it can make the correct decisions when identifying and authenticating users.
	O.TRUSTED_CHANNEL The TOE will provide a means to establish protected communications with remote entities based	O.TRUSTED_CHANNEL mitigates by controlling how the TOE communicates with remote entities. Communication with remote entities is allowed after they have been identified and authenticated by the TOE.

	on the security attributes of the remote entity.	
	OE.PROTECTED_COMMUNICATIONS_CHANNEL Remote authorized IT entities must provide a means to communicate securely with the TOE.	OE.PROTECTED_COMMUNICATIONS_CHANNEL mitigates this threat by requiring remote entities to communicate with the TOE only through secure means. This protects the TOE from unauthorized access to TOE data and resources by requiring the remote entities to authenticate to the TOE before accessing TOE data/resources.
T.POOR_DESIGN Unintentional or intentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a malicious user or program.	O.CHANGE_MANAGEMENT The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.	O.CHANGE_MANAGEMENT plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design. This includes controlling physical access to the TOE's development area, and having an automated configuration management system that ensures changes made to the TOE go through an approval process and only those persons that are authorized can make changes to the TOE's design and its documentation.
	O.SOUND_DESIGN The TOE will be designed using sound design principles and techniques. The TOE design, design principles, and design techniques will be adequately and accurately documented.	O.SOUND_DESIGN counters this threat, to a degree, by requiring that the TOE be developed using sound engineering principles. By accurately and completely documenting the design of the security mechanisms in the TOE, including a security model, the design of the TOE can be better understood, which increases the chances that design errors will be discovered.
	O.VULNERABILITY_ANALYSIS_TEST The TOE will undergo appropriate independent	O.VULNERABILITY_ANALYSIS_TEST ensures that the design of the TOE is independently analyzed for design flaws. Having an independent party perform the assessment ensures an objective approach is taken and may find errors in the design that would be

	vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.	left undiscovered by developers that have a preconceived incorrect understanding of the TOE's design.
T.POOR_IMPLEMENTATION Unintentional or intentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a malicious user or program.	O.CHANGE_MANAGEMENT The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.	O.CHANGE_MANAGEMENT plays a role in mitigating this threat in the same way that the poor design threat is mitigated. By controlling who has access to the TOE's implementation representation and ensuring that changes to the implementation are analyzed and made in a controlled manner, the threat of intentional or unintentional errors being introduced into the implementation are reduced.
	O.SOUND_IMPLEMENTATION The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.	In addition to documenting the design so that implementers have a thorough understanding of the design, O.SOUND_IMPLEMENTATION requires that the developer's tools and techniques for implementing the design are documented. Having accurate and complete documentation, and having the appropriate tools and procedures in the development process helps reduce the likelihood of unintentional errors being introduced into the implementation.
	O.THOROUGH_FUNCTIONAL_TESTING The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional	Although the previous three objectives help minimize the introduction of errors into the implementation, O.THOROUGH_FUNCTIONAL_TESTING increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing.

	requirements.	
	O.VULNERABILITY_ANALYSIS_TEST The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.	O.VULNERABILITY_ANALYSIS_TEST helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation, and the fact that exhaustive testing of the external interfaces is not required, may leave bugs in the implementation undiscovered in functional testing. Having an independent party perform a vulnerability analysis and conduct testing outside the scope of functional testing increases the likelihood of finding errors.
T.POOR_TEST Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.	O.CORRECT_ TSF_OPERATION The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.	While these testing activities are necessary for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded. O.CORRECT_ TSF_OPERATION ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software, including the cryptographic functions) can be demonstrated, and thus providing end users the confidence that the TOE's security policies continue to be enforced.
	O.THOROUGH_FUNCTIONAL_TESTING The TOE will undergo appropriate	Design analysis determines that TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of

	security functional testing that demonstrates the TSF satisfies the security functional requirements.	the TOE's security mechanisms must be performed during the evaluation of the TOE. O.THOROUGH_FUNCTIONAL_ TESTING ensures that adequate functional testing is performed to demonstrate the TSF satisfies the security functional requirements and that the TOE's security mechanisms operate as documented. While functional testing serves an important purpose, it does not ensure the TSFI cannot be used in unintended ways to circumvent the TOE's security policies.
	O.VULNERABILITY_ANALYSIS_TEST The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.	O.VULNERABILITY_ANALYSIS_TEST addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing.
T.REPLAY A user may gain inappropriate access to the TOE by replaying authentication information, or may cause the TOE to be inappropriately configured by replaying TSF data or security attributes (e.g., captured as	O.REPLAY_DETECTION The TOE will provide a means to detect and reject the replay of authentication data as well as other TSF data and security attributes.	O.REPLAY_DETECTION detects a user from replaying authentication data. Detection of replay of authentication data will counter the threat that a user will be able to record an authentication session between a trusted entity (administrative user or trusted IT entity) and then replay it to gain access to the TOE, as well as counter the ability of a user to act as another user.

transmitted during the course of legitimate use).		
T.RESIDUAL_DATA A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. This means that network packets sent in response to a request will not have residual data from another packet (potentially from another user) due to the padding of a packet.
T.RESOURCE_EXHAUSTION A malicious process or user may block others from system or network resources (e.g., network applications) via a resource exhaustion denial of service attack.	O.RESOURCE_SHARING The TOE will provide mechanisms that mitigate attempts to exhaust CPU resources provided by the TOE (e.g., network authentication.).	O.RESOURCE_SHARING mitigates this threat by requiring the TOE to provide controls relating to two different resources: CPU time and available network connections. The administrator is allowed to specify a percentage of processor time that is allowed to be used so that an attempt to exhaust the resource will fail when it reaches the quota. This objective also addresses the denial-of-service attack of a user attempting to exhaust the connection-oriented resources by generating a large number of half-open connections (e.g., SYN attack).
T.SPOOFING A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.	O.TRUSTED_PATH The TOE will provide a means to ensure that users are not communicating with some other entity pretending to be the TOE when supplying identification and authentication data.	It is possible for an entity other than the TOE (a subject on the TOE, or another IT entity on the network between the TOE and the end user) to provide an environment that may lead a user to mistakenly believe they are interacting with the TOE, thereby fooling the user into divulging identification and authentication information. O.TRUSTED_PATH mitigates this threat by ensuring users have the capability to ensure they are communicating with the TOE when providing identification and authentication data to the TOE.

	O.TRUSTED_CHANNEL The TOE will provide a means to establish protected communications with remote entities based on the security attributes of the remote entity.	O.TRUSTED_CHANNEL mitigates this threat similar to O_TRUSTED_PATH by providing remote components the capability to ensure they are communicating with the TOE when providing identification and authentication data to the TOE.
	OE.PROTECTED_COMMUNICATIONS_CHANNEL Remote authorized IT entities must provide a means to communicate securely with the TOE.	OE.PROTECTED_COMMUNICATIONS_CHANNEL is necessary to mitigate this threat because the trusted channel requires a protected communications channel to provide authenticity of the communicating entities, such as the TOE.
T.TSF_COMPROMISE A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION is necessary to mitigate this threat by ensuring no TSF data remain in resources allocated to a user. Even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data.
	O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure.	O.SELF_PROTECTION requires that the TSF be able to protect itself from tampering and that the security mechanisms in the TSF cannot be bypassed. Without this objective, there could be no assurance that users could not view or modify TSF data or TSF executables.
	O.MANAGE	O.MANAGE provides the capability to restrict access to TSF to those that are

	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	authorized to use the functions. Satisfaction of this objective (and its associated requirements) prevents unauthorized access to TSF functions and data through the administrative mechanisms.
	O.DISPLAY_BANNER The TOE will display an advisory warning regarding use of the TOE.	O.DISPLAY_BANNER helps mitigate this threat by providing the Platform Administrator the ability to remove product information (e.g., product name, version number) from a banner that is displayed to users. Having product information about the TOE provides an attacker with information that may increase their ability to compromise the TOE.
	O.TRUSTED_PATH The TOE will provide a means to ensure that users are not communicating with some other entity pretending to be the TOE when supplying identification and authentication data.	O.TRUSTED_PATH plays a role in addressing this threat by ensuring that there is a trusted communication path between the TSF and various users (relying parties (for authentication) and trusted IT entities (for performing replication, for instance)). This ensures the transmitted data cannot be compromised or disclosed during the duration of the trusted path. The protection offered by this objective is limited to TSF data, including authentication data and all data sent or received by trusted IT entities (a relying party's user data is not protected; only the authentication portion of the session is protected).
	O.TRUSTED_CHANNEL The TOE will provide a means to establish protected communications with	O.TRUSTED_CHANNEL helps mitigate the threat of TSF compromise by malicious remote entities by requiring the TOE to identify and authenticate remote entities before communicating with them.

	remote entities based on the security attributes of the remote entity.	
	OE.PROTECTED_COMMUNICATIONS_CHANNEL Remote authorized IT entities must provide a means to communicate securely with the TOE.	OE.PROTECTED_COMMUNICATIONS_CHANNEL is necessary to mitigate this threat because it provides confidentiality, integrity, and authenticity of communications between remote entities and the TOE, thereby preventing inappropriate access to TSF data.
	OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.	OE.PHYSICAL helps mitigate this threat by controlling physical access to the TOE and the TSF, decreasing the opportunities to compromise the TSF.
T.UNATTENDED_SESSION A user may gain unauthorized access to an unattended session.	O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	O. ROBUST_TOE_ACCESS helps to mitigate this threat by including mechanisms that place controls on user's sessions. Local administrator's sessions and remote sessions are locked after an administrator-defined time period of inactivity. Locking the local administrator's session reduces the opportunity of someone gaining unauthorized access the session when the console is unattended. Dropping the connection of a remote session (after the specified time period) reduces the risk of someone accessing the remote machine where the session was established, thus gaining unauthorized access to the session.
T.UNAUTHORIZED_ACCESS	O.MEDIATE	O.MEDIATE works to mitigate this threat by requiring that TOE data is

A user may gain access to user data for which they are not authorized according to the TOE security policy.	The TOE must protect user data in accordance with its security policy.	protected using access control items. An access control item contains information about who is allowed to access an object, as well as the allowed modes of access. The settings present in the access control item selected in the access control decision process determine whether or not a user is authorized to access the object. It should be noted that multiple security policies can be (but do not have to be) in place in a single TOE, meaning that the process by which the target ACI is selected can be different for two different objects. It is required, however, that all objects be covered by this policy. Note that O.SELF_PROTECTION ensures that this access control mechanism is always invoked, thus ensuring that users cannot bypass the mechanism to access data for which they are not authorized.
	O.USER_GUIDANCE The TOE will provide users with the information necessary to correctly use the security mechanisms.	O.USER_GUIDANCE mitigates this threat by providing the user the information necessary to use the security mechanisms that control access to user data in a secure manner.
T.UNIDENTIFIED_ACTIONS The administrator may fail to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.	O.AUDIT_REVIEW The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.	O.AUDIT_REVIEW helps to mitigate this threat by providing a variety of mechanisms for monitoring the use of the system. The audit review is performed through analysis of the audit trail produced by the audit mechanism. For analyzing the audit trail, the TOE requires an Auditor role. This role is restricted to Audit record review and the deletion of the audit trail for maintenance purposes. The TOE's audit analysis mechanism must consist of a minimum set of configurable audit events that could

		indicate a potential security violation. Thresholds for these events must be configurable by an appropriate administrative role. By configuring these auditable events, the TOE monitors the occurrences of these events (e.g. set number of authentication failures, self-test failures, etc.) and immediately notifies an administrator once an event has occurred or a set threshold has been met. If a potential security violation has been detected, the TOE displays a message that identifies the potential security violation to the administrative console. This message is displayed and will remain on the screen until an administrator acknowledges the message. At this point, the administrator will receive notification that the alarm has been acknowledged, who acknowledged the alarm, and the time that it was acknowledged. In addition to displaying the potential security violation, the message must contain all audit records that generated the potential security violation. By enforcing the message content and display, this objective provides assurance that a TOE administrator will be notified of a potential security violation.
T.UNKNOWN_STATE When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.	O.MAINT_MODE The TOE will provide a mode from which recovery or initial startup procedures can be performed.	O.MAINT_MODE helps to mitigate this threat by ensuring that the TOE does not continue to operate in an insecure state when a hardware or software failure occurs. After a failure, the TOE enters a state that disallows operations and requires an administrator to follow documented procedures to return the TOE to a secure state.
	O.CORRECT_ TSF_OPERATION	O.CORRECT_TSF_OPERATION counters this threat by ensuring that the TSF runs a suite of tests to successfully

	The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.	demonstrate the correct operation of the TSF (hardware and software) and the TSF's cryptographic components at initial startup of the TOE. In addition to ensuring that the TOE's security state can be verified, an administrator can verify the integrity of the TSF's data and stored code as well as the TSF's cryptographic data and stored code using the TOE-provided cryptographic mechanisms.
	O.SOUND_DESIGN The TOE will be designed using sound design principles, and techniques. The TOE design, design principles and design techniques will be adequately and accurately documented.	O.SOUND_DESIGN works to mitigate this threat by requiring that the TOE developers provide accurate and complete design documentation of the security mechanisms in the TOE, including a security model. By providing this documentation, the possible secure states of the TOE are described, thus enabling the administrator to return the TOE to one of these states during the recovery process.
	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management of the TOE.	O. ROBUST_ADMIN_GUIDANCE provides administrative guidance for the secure start-up of the TOE as well as guidance to configure and administer the TOE securely. This guidance provides administrators with the information necessary to ensure that the TOE is started and initialized in a secure manor. The guidance also provides information about the corrective measure necessary when a failure occurs (i.e., how to bring the TOE back into a secure state).

Table 8. Threat to Objective Mapping

The following table shows what objectives address a particular policy and how the objectives apply to the policy.

Policy	**Objective Addressing Policy**	**Rationale**
P.ACCESS_BANNER The TOE shall display an initial banner	O.DISPLAY_BANNER The TOE will display an advisory warning regarding	O.DISPLAY_BANNER satisfies this policy by ensuring that the TOE displays an

describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.	use of the TOE.	Administrator-configurable banner that provides all users with a warning about the unauthorized use of the TOE. This is required to be displayed before an interactive administrative session, since it does not make sense to display a banner for sessions involving authentication requests from users, and those types of sessions are largely automated.
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security-relevant events associated with users.	O.AUDIT_GENERATION addresses this policy by providing an audit mechanism to record the actions of a specific user, as well as the capability for an administrator to "pre-select" audit events based on the user ID. The audit event selection function is configurable during run-time to ensure the TOE is able to capture security-relevant events given changes in threat conditions. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE (e.g. access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.). Attributes used in the audit record generation process are also required to be bound to the subject, ensuring users are held accountable
	O.TIME_STAMPS The TOE will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.	O.TIME_STAMPS plays a role in supporting this policy by requiring the TOE to provide a reliable time stamp (configured locally by the Administrator or via a trusted IT entity, such as an NTP

		server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID will also include the date and time that the event occurred.
	O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	O. ROBUST_TOE_ACCESS supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.
P.LOCAL_ADMIN_ACCESS Administrators shall be able to administer the TOE locally only.	O.ADMIN_ROLE The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally.	O.ADMIN_ROLE supports this policy by requiring the TOE to provide mechanisms that allow local administration of the TOE.
	O.TRUSTED_PATH The TOE will provide a means to ensure that users are not communicating with some other entity pretending to be the TOE when supplying identification and authentication data.	O.TRUSTED_PATH satisfies this policy by requiring that each remote administrative and management session for all trusted users is authenticated and conducted via a secure channel. Additionally, all trusted IT entities (e.g., trusted Application Management Servers, Trusted Path Extensions) connect through a protected channel, thus avoiding disclosure and spoofing problems.
P.CRYPTOGRAPHY The TOE shall use NIST FIPS validated cryptography as a baseline with additional NSA-approved methods for key management (i.e.; generation, access,	O.CRYPTOGRAPHY The TOE will use NIST FIPS 140-2 validated cryptographic services.	O.CRYPTOGRAPHY implements this policy by requiring the TOE to implement NIST FIPS-validated cryptographic services. The objective requires that the functions needed by the TOE are FIPS approved, and further that

distribution, destruction, handling, and storage of keys), and for cryptographic operations (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services).		they are available in a FIPS-approved mode of operation of the cryptomodule.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION implements this policy by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. This means that network packets sent in response to a request will not have residual data from another packet (potentially from another user) due to the padding of a packet.
P.SINGLE_SIGN_ON Authorized users shall be able to access services on a federation of servers after successful authentication.	O.SINGLE_SIGN_ON The TOE will provide a means to ensure users will be able to access services on a federation of servers after successful authentication.	O.SINGLE_SIGN_ON implements this policy by providing the mechanisms to establish single sign-on. This includes authenticating the user through an authorized TPE, maintaining the Application Mapping Database (AMDB) and distributing subsets of the AMDBs (that map applications to the hosting servers) to the appropriate TPEs, maintaining the User Database (containing currently authenticated users), and distributing the information in the User Database to authorized AMSes.
	O.SINGLE_SIGN_ON_SUPPORT The TOE will provide either centralized or distributed user identification and authentication mechanisms	O.SINGLE_SIGN_ON_SUPPORT supports this policy by requiring the TOE to perform the single sign-on mechanisms by the TOE itself (central SSO) or by distributing the functions

	that are secure.	across multiple TOEs (for increased performance and availability, and/or other reasons).
P.VULNERABILITY_ ANALYSIS_TEST The TOE shall undergo appropriate independent vulnerability analysis and penetration testing to demonstrate that the TOE is resistant to an attacker possessing a medium attack potential.	O.VULNERABILITY_ANALYSIS_ TEST The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.	O.VULNERABILITY_ANALYSIS_TEST satisfies this policy by ensuring that an independent analysis is performed on the TOE and penetration testing based on that analysis is performed. Having an independent party perform the analysis helps ensure objectivity and eliminates preconceived notions of the TOE's design and implementation that may otherwise affect the thoroughness of the analysis. The level of analysis and testing requires that an attacker with a moderate attack potential cannot compromise the TOE's ability to enforce its security policies.

Table 9. Policy to Objective Mapping

E. ASSUMPTION MAPPING

The following table shows how the objectives of the environment address the assumptions.

Assumption	Environment Objectives Addressing Assumption	Rationale
A.PHYSICAL It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.	OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.	OE.PHYSICAL addresses this assumption by requiring that physical security, commensurate with the value of the TOE and the data it contains, be provided by the environment.
A.REMOTE_COMPONENT_ENVIRONMENT	OE.REMOTE_COMPONENT_ENVIRONMENT	OE.REMOTE_COMPONENT_ENVIRONMENT addresses this assumption by

The accreditation process will ensure that the procuring organization will manage and protect the remote components in a manner that is commensurate with the protection mechanisms provided by the TOE.	The accreditation process will ensure that the procuring organization will manage and protect the remote components (i.e., Application Management Servers, Trusted Path Extensions) in a manner that is commensurate with protection provided for the TOE.	requiring sufficient documentation and evidence from the certification and accreditation process that the remote components that interact with the TOE be managed and protected in a manner that is commensurate with protection provided by the TOE.
A.REMOTE_COMPONENT_FUNCTIONALITY Remote component processes that interact with the TOE are trusted to comply with the security requirements levied upon them by the TOE.	OE.REMOTE_COMPONENT_FUNCTIONALITY Remote component trusted processes will be constructed to comply with the security requirements levied upon them by the TOE.	OE.REMOTE_COMPONENT_FUNCTIONALITY addresses this assumption by requiring remote component processes to be soundly constructed in order to comply with the TOE's security requirements.
A.NETWORK_SECURITY_POLICY_ENFORCEMENT All network components are vetted with the appropriate level of trust in order to properly enforce the network security policy.	OE.NETWORK_SECURITY_POLICY_ENFORCEMENT The security administrator must ensure that the appropriate level of trust has been established among all components such that network security policies are understood and enforced.	OE.NETWORK_SECURITY_POLICY_ENFORCEMENT satisfies this assumption by requiring the administration of all network components (i.e., the TOE, Application Management Servers, and Trusted Path Extensions) is properly performed such that the network security policy can be correctly enforced.

Table 10. Assumption to Environmental Objective Mapping

F. REQUIREMENTS MAPPING

The objectives for the Authentication Server need to be mapped to requirements that will implement these objectives. As a reminder, the terms TOE and TSF refer to the Authentication Server (AUS). The objectives use the term TOE and TSF; in contrast, the requirements use the term AUS because the objectives closely follow Common Criteria conventions while the requirements do not (allows for more flexibility in constructing the requirements). The numbers preceding each requirement refers to the section of this chapter where the requirement can be found.

Objectives to Requirements Mapping
O.ADMIN_ROLE: The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally.
B.8.2 The AUS shall maintain the roles: Security Administrator; Cryptographic Administrator (i.e., users authorized to perform cryptographic initialization and management functions); and Audit Administrator. B.8.3 The AUS shall be able to associate users with roles.
O.AUDIT_GENERATION: The TOE will provide the capability to detect and create records of security-relevant events associated with users.
B.1.1 The Authentication Server (AUS) shall have configurable auditing capabilities. The levels of auditing are hierarchical, from the least amount of audit information to the most. The Authentication Server will support the following audit levels (from high to low): alert, critical, error, warning, notice, information, and debugging. All audited events shall be recorded. B.1.2 The types of events that shall be audited include user authentication, Application Management Server (AMS) retrieval of user authentication information, transfer of an Application Mapping Database to a remote device, user session level change/logout, failure in a remote component, and reading/modification of the audit trail. B.1.3 The date and time of the event, IP address of the remote host, type of event, user name (if applicable), user session level (if applicable), and the event outcome (success or failure) shall be recorded. B.1.7. The audit records generated by the AUS shall be in a format that can be parsed. B.5.1 The AUS shall ensure that users are identified and authenticated in order to associate them with the proper security attributes, such as user name and session level, prior to access to AUS data or network applications. B.6.5 The AUS shall be able to provide reliable time stamps for its own use.
O.AUDIT_PROTECTION: The TOE will provide the capability to protect audit

information by controlling access to the audit trail.
B.1.6 Authorized administrators shall be the only entities able to review, delete, or modify the audit logs. Authorized administrators shall be able to configure the actions to take when an audit log is full.
O.AUDIT_REVIEW: The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.
B.1.4 The AUS shall alarm the Security Administrator whenever a security violation has been violated through a message displayed on the local console identifying the violation and allow access to the audit records associated with the event. B.1.5 The AUS shall display an acknowledgement message identifying a reference to the potential security violation, a notice that it has been acknowledged, the time of the acknowledgement and the user identifier that acknowledged the alarm, at the local console.
O.CHANGE_MANAGEMENT: The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.
C.1.1 The software and documentation (e.g., design specifications, guidance documents) for the Authentication Server, including configuration files, shall be placed under configuration management (CM). C.1.2 The CM system will uniquely identify configuration items, including those associated with the AUS (i.e., Authentication Server implementation and documentation). C.1.3 The CM documentation shall include a configuration list, a CM plan, and an acceptance plan. The CM documentation shall provide evidence that the CM is maintaining the configuration items. C.1.4. The acceptance plan shall describe the acceptance procedures for modified or newly created configuration items of the AUS. C.5.1 The AUS shall follow the same life cycle model and developmental procedures as the MYSEA project.

O.CORRECT_ TSF_OPERATION: The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.
B.6.7 The AUS shall run a suite of self-tests during initial start-up, periodically during normal operation as specified by an authorized administrator, and at the request of an authorized administrator to demonstrate the correct operation of the software portions of the AUS.
O.CRYPTOGRAPHY: The TOE will use NIST FIPS 140-2 validated cryptographic services.
B.3.1 The AUS shall use NIST-validated cryptographic standards when using cryptography for communications to remote IT entities. B.3.2 The cryptographic keys used by the AUS shall be managed using NIST-validated mechanisms. This includes the generation and destruction of cryptographic keys. B.3.3 The AUS shall use NIST-validated cryptomodules in a NIST-validated mode for cryptographic operations.
O.DISPLAY_BANNER: The TOE will display an advisory warning regarding use of the TOE.
B.9.1 Before establishing a user session that requires authentication, the AUS shall display only an authorized administrator-specified advisory notice and consent warning message regarding unauthorized use of the AUS.
O.MAINT_MODE: The TOE will provide a mode from which recovery or initial startup procedures can be performed.
B.6.6 When automated recovery from failures/service discontinuities is not possible, the AUS shall enter a maintenance mode where the ability to return to a secure state is provided.
O.MANAGE: The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.

B.8.1 The AUS shall restrict configuration of security services management, such as setting quota limits and audit configuration, only to authorized administrators.
O.MEDIATE: The TOE must protect user data in accordance with its security policy.
B.4.1 For all user accesses to AUS resources, the AUS shall enforce the access control policy and information flow policy based on the user ID and session level. B.4.2 For all remote device accesses to AUS resources, the AUS shall enforce the access control policy and information flow policy based on the remote device ID (i.e., TPE ID, AMS ID) and security level.
O.PROTECT_IN_TRANSIT: The TSF will protect user and TSF data when it is in transit from the TOE to another remote entity.
B.2.1 The AUS shall employ cryptographic functionality for secure connection between the AUS and remote IT entities (i.e., TPEs and AMSes).
O.REPLAY_DETECTION: The TOE will provide a means to detect and reject the replay of authentication data as well as other TSF data and security attributes.
B.6.4 The Authentication Server shall detect replay of authentication information, other AUS data, or AUS security attributes transmitted during the course of legitimate use.
O.RESIDUAL_INFORMATION: The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.
B.4.3 The AUS shall ensure that any previous information content of a resource is made unavailable upon the resource's reallocation to any AUS objects.
O.RESOURCE_SHARING: The TOE will provide mechanisms that mitigate attempts to exhaust CPU resources provided by the TOE (e.g., network authentication.).
B.7.1 The AUS shall enforce administrator-specified maximum quotas of the AUS services (i.e., authentication, access to AUS databases) that users and remote entities can use over an administrator-specified period of time.
O. ROBUST_ADMIN_GUIDANCE: The TOE will provide administrators with the necessary information for secure delivery and management of the TOE.

C.2.1 The installation, generation and start-up documentation shall be provided and shall describe all the steps necessary for secure installation, generation and start-up of the AUS. C.4.3 The administrative guidance shall describe the procedures and technical measures necessary to restrict physical access to the system. C.4.4 The administrative guidance shall cover configuration, maintenance, and administration of the Authentication Server in a secure manner. The guidance is intended to help administrators understand the security functions of the Authentication Server, including both those functions that require the administrator to perform security-critical actions and those functions that provide security-critical information to the administrator C.4.5 The administrative guidance shall describe the functions and interfaces available to the administrator in addition to how to manage the AUS in a secure manner. C.4.6 The administrative guidance shall describe all security requirements for the operational environment that are relevant to the administrator and the AUS. C.7.1 Guidance documentation for the Authentication Server shall be complete, clear, consistent, and reasonable. The guidance documentation shall: identify all possible modes of operation of the AUS (including operation following failure or operational error), their consequences and implications for maintaining secure operation; list all assumptions about the intended environment; list all requirements for external security measures (including external procedural, physical and personnel controls); and demonstrate that the guidance documentation is complete.
O. ROBUST_TOE_ACCESS: The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.
B.5.1 The AUS shall ensure that users are identified and authenticated in order to associate them with the proper security attributes, such as user name and session level, prior to access to AUS data or network applications. B.5.2 The AUS shall authenticate registered users based on the user ID and a password.

B.5.3 The AUS shall ensure that Application Management Servers are identified and authenticated prior to access to AUS data. B.5.4 The AUS shall authenticate registered Application Management Servers based on a digital certificate. B.5.5 The AUS shall associate the following user security attributes with subjects acting on the behalf of that user: username, session level, and any other appropriate security attributes. B.5.6 The AUS shall associate the following security attributes with subjects acting on the behalf of the remote component: hostname, host IP address, and any other appropriate security attributes. B.5.7 The AUS shall detect when an administrator-configurable number of unsuccessful authentication attempts occur within an administrator-configurable time period. B.9.2 The AUS shall be able to deny session establishment based on the TPE ID, user ID, and user clearance. B.9.3 The AUS shall provide AUS-initiated session locking after an administratively-set timeout interval.
O.SELF_PROTECTION: The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure.
B.6.1 The Authentication Server shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects. B.6.2 The Authentication Server shall enforce separation between the security domains of subjects in the AUS scope of control. B.6.8 The AUS shall ensure that security policy enforcement functions are invoked and succeed before each function within the AUS scope of control is allowed to proceed.
O.SINGLE_SIGN_ON: The TOE will provide a means to ensure users will be able to access services on a federation of servers after successful authentication.

B.6.3 The AUS shall ensure the availability of user session information provided to a remote component within a security administrator-configurable time given the remote component has been authenticated to the AUS and is functioning normally. B.11.1 The AUS shall maintain a user database containing information about currently authenticated users (e.g., user ID, session level, TPE ID, etc). B.11.2 The AUS shall distribute information in the user database to authorized, authenticated remote entities (i.e., AMSes) when user authentication information is requested by the remote entity and when a user's session status changes (e.g., user logout or session level change). B.11.3 The AUS shall maintain an Application Mapping Database that maps applications to hosting Application Management Servers. B.11.4 The AUS shall distribute portions of the Application Mapping Database to authorized remote entities (i.e., TPEs) after successful user authentication, successful user session negotiation, and upon SAK press during a user's session when the AMDB subset on the TPE is inconsistent with the AUS's AMDB.
O.SINGLE_SIGN_ON_SUPPORT: The TOE will provide either centralized or distributed user identification and authentication mechanisms that are secure.
B.11.5. The AUS shall be able to provide single sign-on services whether it acts alone or with multiple AUSes. B.11.6. The AUS shall be able to securely distribute single sign-on management information (e.g., user database, Application Mapping Database) to authorized AUSes.
O.SOUND_DESIGN: The TOE will be designed using sound design principles, and techniques. The TOE design, design principles and design techniques will be adequately and accurately documented.
C.3.1 An informal functional specification describing the interfaces of the security functions for the Authentication Server shall be provided. C.3.2 An informal high-level design of the AUS shall be developed.

C.3.3 An informal architectural design of the AUS security functions shall be developed in detail sufficient to determine that the security enforcing mechanisms cannot be bypassed. C.3.4 An informal low level design shall be developed for the AUS. C.3.5 The design of the AUS shall meet the functional requirements. C.3.8 An informal security policy model of the AUS shall be developed.
O.SOUND_IMPLEMENTATION: The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.
C.3.6. The implementation of the AUS shall be an accurate instantiation of the design. C.3.7 The implementation of the AUS shall be adequately and accurately documented such that the AUS can be generated without further design decisions.
O.THOROUGH_FUNCTIONAL_ TESTING The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.
C.6.1 The AUS and its security functions shall be tested to ensure that it operates in accordance with its high-level design and low-level design. C.6.2 Test documentation (e.g., test plan, procedures, and results) shall be produced.
O.TIME_STAMPS: The TOE will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.
B.6.5 The AUS shall be able to provide reliable time stamps for its own use. B.8.1 The AUS shall restrict configuration of security services management, such as setting quota limits and audit configuration, only to authorized administrators.
O.TRUSTED_CHANNEL: The TOE will provide a means to establish protected communications with remote entities based on the security attributes of the remote entity.
B.10.1 The AUS shall provide a communication channel between itself and remote components that is logically distinct from other communication channels and provides

assured identification of its end points and protection of the channel data from modification or disclosure. B.10.2 The AUS shall permit itself or an authorized remote component initiate communication via the trusted channel for remote user/component authentication, SSO service management, and other network security management functions.
O.TRUSTED_PATH: The TOE will provide a means to ensure that users are not communicating with some other entity pretending to be the TOE when supplying identification and authentication data.
B.10.3 The AUS shall provide a communication path between itself and local users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure. B.10.4 The AUS shall permit local users to initiate communication via the trusted path for initial user authentication and session negotiation.
O.USER_GUIDANCE: The TOE will provide users with the information necessary to correctly use the security mechanisms.
C.4.1 The user guidance shall describe the interaction between the user and the Authentication Server for proper authentication, session level modification, and logout. C.4.2 The user guidance shall clearly present all user responsibilities necessary for secure use of the Authentication Server.
O.VULNERABILITY_ANALYSIS_ TEST: The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.
C.7.2 The developer shall perform a vulnerability analysis and provide vulnerability analysis documentation.

Table 11. Objectives to Requirements Mapping

G. SUMMARY

This chapter described the initial set of functional and assurance security requirements for the Authentication Server. The threats, policies, and assumptions were mapped to objectives, both system and environmental. These objectives were then mapped back to the security requirements, transitively demonstrating how the requirements address the threats, policies, and assumptions applicable to the Authentication Server.

V. FUTURE WORK AND CONCLUSIONS

A. INTRODUCTION

This chapter presents future work and the conclusions for the MYSEA single sign-on project. This thesis laid out the initial SSO design and requirements; they will both be further refined in future iterations of this project. Some suggestions for future work are discussed below.

B. FUTURE WORK

1. Additional Requirements

Because the security requirements in Chapter IV are preliminary in nature, continuing work in revising and refining these requirements is needed. The initial security requirements were specified only for the Authentication Server, but the security requirements for the environment must also be specified. This includes the requirements for the Protected Communications Channel, Trusted Path Extension, and Application Management Server such that single sign-on functionality is supported by these components.

Additional requirements also need to be specified for the Service Management mechanisms that support single sign-on, such as the distribution of the Application Mapping Database (AMDB) and ensuring its consistency across all the TPEs. Other requirements for Service Management that need to be specified are those that monitor the liveliness of the TPEs, AMSes, and the AUS. The requirements to confirm the AMS is hosting the applications as designated in the AMDB also need to be specified.

This thesis assumed only a single Authentication Server to provide the initial user authentication and subsequent single sign-on. However, the objective specified in Chapter III, O.SINGLE_SIGN_ON_SUPPORT allows the AUS to be distributed among multiple machines for increased performance and reliability. Using multiple AUSes for distributed authentication and sign-on requires specification of the requirements for secure distribution of authentication data among the multiple AUSes, requirements for the AUSes to maintain consistency of the authentication data, and load-balancing and reliability requirements for the AUSes.

The analysis in Chapter III (threats, assumptions, policies and objectives) and the requirements in Chapter IV were based on the Common Criteria v.2.2. However, Common Criteria version 3.0 was released late into this thesis so the new standards have not been incorporated in this work. Future work involves integrating Common Criteria version 3.0 elements with the current requirements.

2. Prospective Design Work

This work presented a very high-level design of the overall SSO system for the MYSEA environment in Chapter III. An initial set of requirements was specified in Chapter IV, a complete set of requirement needs to be specified first before work on the low level design can proceed. Some aspects of the SSO system that need to be further designed are the mechanisms for the construction of the Application Mapping Databases, network and application-level monitoring, failure recovery (especially when an AUS fails), and distributed Authentication Server capabilities.

The Protected Communications Channel (PCC) protocol also needs to be designed since all communications between components on the MLS LAN employ the PCC and depend on the PCC to provide communications security. In [5], the high level requirements for the PCC protocol were specified, but this work needs to be reassessed to determine if the protocol is sufficient to support single sign-on. If the existing PCC protocol does not adequately support SSO, enhancements to the PCC protocol must be made to accommodate SSO requirements. Only then can a low level design of the PCC protocol be developed, from which an implementation of the PCC protocol can then be constructed. Other future work may involve moving the AMS-AUS communications to a separate LAN that may or may not require the PCC, but a thorough analysis and design is required.

At a more fine-grained level, the various processes on the AUS and AMSes need to be modified to accommodate single sign-on functionality and management. The existing Trusted Path Server (TPS) and Secure Session Server (SSS) processes on the MLS MYSEA Server will need to be redesigned to support SSO. In particular, the design of how each process will handle user identification and authentication (e.g., querying the AUS, storing received user authentication information from the AUS, and dealing with user session updates from the AUS) requires extensive work.

Also to be modified or designed are the databases used by the AUS and AMS for SSO support. The User Database on the AUS and AMS needs to be redesigned to keep track of the AMSes accessed by a user during a session. The Application Mapping Database maintained by the AUS needs to be designed, as well as a database of valid AMSes that are allowed to query the AUS for user session information. If multiple AUSes are involved, a database specifying which AUS the AMS should query for a user's information needs to be designed. A similar database would also be required for the TPE, so that the TPE knows which AUS to contact for authentication and session services and which AUSes to use in case of the failure of its assigned AUS.

C. CONCLUSIONS

Existing single sign-on solutions were studied and analyzed across a number of security and performance criteria. As a result of this analysis, it was determined that a different kind of single sign-on solution would be constructed for the MYSEA environment. A high level design of the MYSEA SSO solution was developed, which involved designing a new MYSEA architecture and the various CONOPS scenarios to illustrate how the MYSEA components support SSO. The new architecture separated the authentication and application service mechanisms on the MYSEA Server, resulting in the creation of two separate entities: the Authentication Server and the Application Management Server. Because the Authentication Server is the central component in providing single sign-on functionality, the AUS was the focus of study for the development of the security requirements for the SSO system.

The requirements development process informally followed the Common Criteria methodology. A threat analysis was performed on the AUS, and the environmental assumptions and organizational policies were established for the AUS. From there, a set of objectives for the AUS was determined and these objectives were used to construct an initial set of security requirements for the AUS. The security requirements were divided into functional and assurance requirements; these requirements were mapped back to the objectives, which had been mapped to the threats, policies, and assumptions. The results of this work will serve as a framework for future design and specification of single sign-on mechanisms in the MYSEA environment.

Single sign-on greatly enhances the usability of the MYSEA environment by allowing users to authenticate once to access applications on multiple machines. Such capabilities may lead to greater acceptance of MYSEA solutions in the military and intelligence community. The MYSEA single sign-on solution may even be applicable to the DoD's vision of the Global Information Grid (GiG), where users are able to simultaneously access information at multiple security levels across different organizations.

LIST OF REFERENCES

1. Cynthia E. Irvine, Timothy E. Levin, Thuy D. Nguyen, David Shifflett, Jean Khosalim, Paul C. Clark, Albert Wong, Francis Afinidad, David Bibighaus, Joseph Sears, "Overview of a High Assurance Architecture for Distributed Multilevel Security," *Proceedings of the 5th IEEE Systems, Man and Cybernetics Information Assurance Workshop*, United States Military Academy, West Point, NY, pg 38-45, June 10-11, 2004.

2. D. E. Bell and L. LaPadula, "Secure Computer System: Unified Exposition and Multics Interpretation," Mitre Corp., Hanscomb AFB, MA, Tech Rep ESD-TR-76-372, 1975.

3. K. J. Biba, "Integrity Considerations for Secure Computer Systems," MITRE Corp., Tech. Rep. ESD-TR-76-372, 1977.

4. Robert C. Cooper, "Remote Application Support in a Multilevel Environment," Master's thesis, Naval Postgraduate School, Monterey, California, 2005.

5. Joseph D. Sears, "Simultaneous Connection Management and Protection in a Distributed Multilevel Security Environment," Master's thesis, Naval Postgraduate School, Monterey, California, 2004.

6. John F. Horn, "IPsec-Based Dynamic Security Services for the MYSEA Environment," Master's thesis, Naval Postgraduate School, Monterey, California, 2005.

7. Jan De Clercq, "Single Sign-on Architectures," *Proceedings of Infrastructure Security: International Conference, InfraSec 2002*, Bristol, UK, pg 40-58, October 1-3, 2002.

8. B. Clifford Neuman and Theodore Ts'o. "Kerberos: An Authentication Service for Computer Networks," *IEEE Communications*, Vol. 32, No. 9, pg 33-38, September 1994.

9. P. V. McMahon, "SESAME V2 Public Key and Authorisation Extensions to Kerberos," *Proceedings of the 1995 Symposium on Network and Distributed System Security (SNDSS'95)*, Washington, D.C., pg 114-131, February 16-17, 1995.

10. Novell, Inc., "Novell Nsure SecureLogin 3.5: Security Identity Management," technical white paper, 2003. Available: http://www.novell.com/collateral/4621348/4621348.pdf. Accessed: April 2005.

11. Microsoft, “Microsoft .NET Passport Review Guide,” technical white paper, January 2004. Available: http://download.microsoft.com/download/a/f/4/af49b391-086e-4aa2-a84b-ef6d916b2f08/passport_reviewguide.doc. Accessed: August 2004.

12. Common Criteria Project Sponsoring Organizations, "Common Criteria for Information Technology Security Evaluation," CCIMB-2005-07-002, Version 3.0, July 2005. Available: http://niap.nist.gov/cc-scheme/cc_docs/index.html. Accessed: August 2005.

13. Common Criteria Project Sponsoring Organizations, "Common Criteria for Information Technology Security Evaluation," CCIMB-2004-01-001, Version 2.2, January 2004. Available: http://niap.nist.gov/cc-scheme/cc_docs/index.html. Accessed: May 2005.

14. National Security Agency, "U.S. Government Directory Protection Profile for Medium Robustness Environments," Version 1.0, September 1, 2004. Available: http://niap.nist.gov/cc-scheme/pp/PP_VID1031-PP.pdf. Accessed: August 2005.

15. National Security Agency, “Consistency Instruction Manual for Development of US Government Protection Profiles for Use in Medium Robustness Environments,” Version 3.0, February 1, 2005. Available: http://niap.nist.gov/cc-scheme/cc_docs/cem_v12.pdf. Accessed: June 2005.

16. Douglas R. Kane Jr., “Web-Based Dissemination System for the Trusted Computing Exemplar Project,” Master’s thesis, Naval Postgraduate School, Monterey, California, 2005.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Hugo A. Badillo
   NSA
   Fort Meade, MD

4. George Bieber
   OSD
   Washington, DC

5. RADM Joseph Burns
   Fort George Meade, MD

6. John Campbell
   National Security Agency
   Fort Meade, MD

7. Deborah Cooper
   DC Associates, LLC
   Roslyn, VA

8. CDR Daniel L. Currie
   PMW 161
   San Diego, CA

9. Louise Davidson
   National Geospatial Agency
   Bethesda, MD

10. Vincent J. DiMaria
    National Security Agency
    Fort Meade, MD

11. LCDR James Downey
    NAVSEA
    Washington, DC

12. Dr. Diana Gant
National Science Foundation

13. Jennifer Guild
SPAWAR
Charleston, SC

14. Richard Hale
DISA
Falls Church, VA

15. LCDR Scott D. Heller
SPAWAR
San Diego, CA

16. Wiley Jones
OSD
Washington, DC

17. Russell Jones
N641
Arlington, VA

18. David Ladd
Microsoft Corporation
Redmond, WA

19. Dr. Carl Landwehr
National Science Foundation
Arlington, VA

20. Steve LaFountain
NSA
Fort Meade, MD

21. Dr. Greg Larson
IDA
Alexandria, VA

22. Penny Lehtola
NSA
Fort Meade, MD

23. Ernest Lucier
Federal Aviation Administration
Washington, DC

24. CAPT Deborah McGhee
Headquarters U.S. Navy
Arlington, VA

25. Dr. Vic Maconachy
NSA
Fort Meade, MD

26. Doug Maughan
Department of Homeland Security
Washington, DC

27. Dr. John Monastra
Aerospace Corporation
Chantilly, VA

28. John Mildner
SPAWAR
Charleston, SC

29. Jim Roberts
Central Intelligence Agency
Reston, VA

30. Keith Schwalm
Good Harbor Consulting, LLC
Washington, DC

31. Charles Sherupski
Sherassoc
Round Hill, VA

32. Dr. Ralph Wachter
ONR
Arlington, VA

33. David Wirth
N641
Arlington, VA

34. Daniel Wolf
NSA
Fort Meade, MD

35. Jim Yerovi
NRO
Chantilly, VA

36. CAPT Robert Zellmann
CNO Staff N614
Arlington, VA

37. Dr. Cynthia E. Irvine
Naval Postgraduate School
Monterey, CA

38. Thuy D. Nguyen
Naval Postgraduate School
Monterey, CA

39. Timothy E. Levin
Naval Postgraduate School
Monterey, CA

40. Sonia Bui
Civilian, Naval Postgraduate School
Monterey, CA