

A User Comment Draft of the Joint Committee on the NTCIP

NTCIP 1105 v01.02

National Transportation Communications for ITS Protocol CORBA Security Service Specification

April 20, 2001; based on v01.01 of August 24, 2000

This is a draft document, which is distributed for review and comment purposes only. You may reproduce and distribute this document within your organization, but only for the purposes of and only to the extent necessary to facilitate review and comment to the **NTCIP Coordinator**. Please ensure that all copies reproduced or distributed bear this legend. This document contains preliminary information which is subject to change.

Published by

American Association of State Highway and Transportation Officials (AASHTO)

444 North Capitol Street, N.W., Suite 249
Washington, D.C. 20001

Institute of Transportation Engineers (ITE)

1099 14th Street, N.W., Suite 300 West
Washington, D.C. 20005-3438

National Electrical Manufacturers Association (NEMA)

1300 North 17th Street, Suite 1847
Rosslyn, Virginia 22209-3801

ACKNOWLEDGEMENTS

This publication was prepared by the NTCIP Center-to-Center Working Group, a subdivision of the Joint Committee on the NTCIP. The Joint Committee is organized under a Memorandum of Understanding among the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). The NTCIP development effort is guided by the Joint Committee on the NTCIP, which consists of six representatives from each of the above organizations.

At the time that this document was prepared, the following individuals were active members of the NTCIP Center-to-Center Working Group:

- Ken Vaughn
- Warren Tighe
- Greg Mosley
- Walt Townsend
- Allan Foodym
- Bob Rausch
- Ray Starr
- Jeff Mayo
- Steve Dellenback
- Jeff Brummond
- Blake Christie
- Scott Melby
- Paul Olson

In addition to the many volunteer efforts, recognition is also given to those organizations who supported the efforts of the working groups by providing comments and funding for the standard, including:

- ADF Consulting
- Federal Highway Administration
- Gardner Systems
- Iteris
- Minnesota Department of Transportation
- MitreTek Systems
- National Engineering Technologies
- PB Farradyne
- Southwest Research Institute
- Texas Department of Transportation
- TransCore
- Washington State Department of Transportation

FOREWORD

This document uses only metric units.

This publication defines the CORBA Security Service Specification for use in center-to-center communications in the transportation domain. There are no annexes to this document.

This document is an NTCIP Base Standard. Base Standards provide formal definitions of protocols or services upon which higher level applications are dependent for use within NTCIP systems.

For more information about NTCIP standards, visit the NTCIP Web Site at <http://www.ntcip.org>. For a hardcopy summary of NTCIP information, contact the NTCIP Coordinator at the address below.

In preparation of this NTCIP document, input of users and other interested parties was sought and evaluated. Inquires, comments, and proposed or recommended revisions should be submitted to:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1847
Rosslyn, Virginia 22209-3801
fax: (703) 841-3331
e-mail: ntcip@nema.org

INTRODUCTION

This publication provides definitions of the CORBA Security Service Specification for use in center-to-center communications in the transportation domain.

This standard defines requirements that are applicable to NTCIP environments that use CORBA.

The following keywords apply to this document: AASHTO, ITE, NEMA, NTCIP, CORBA.

The effort to develop NTCIP began in 1992 with the 3-TS Transportation Management Systems and Associated Control Devices Section of NEMA. Their original desire was to address a user need for extending the TS 2 Standard for traffic control hardware to include standardized systems communication. Under the guidance of the Federal Highway Administration's (FHWA) NTCIP Steering Group, the NEMA effort was expanded to include the development of communications standards for all transportation field devices that could be used in an Intelligent Transportation Systems (ITS) network and for center-to-center communications.

In September 1996, a formal agreement was reached among NEMA, ITE, and AASHTO to jointly develop, approve, and maintain NTCIP Standards. The Center-to-Center Working Group commenced work in 1996.

CONTENTS

BACKGROUND	VI
SECTION 1 GENERAL	1-1
1.1 Scope.....	1-1
1.2 Protocol – Layer Relationship.....	1-1
1.3 References	1-1
1.3.1 Normative References.....	1-2
1.3.2 Informative References.....	1-3
1.4 Definitions, Acronyms, and Terms	1-3
SECTION 2 REQUIREMENTS AND CONFORMANCE	2-1
2.1 General Requirements	2-1
2.2 Functional Requirements.....	2-1
2.2.1 Administrator Actor	2-2
2.2.2 User Actor	2-2
2.2.3 Filter User Access Use Case.....	2-2
2.2.4 Encrypt Communication Use Case.....	2-3
2.2.5 Report User Information.....	2-3
2.2.6 Store User Password Use Case	2-3
2.2.7 Input User Information Use Case	2-3
2.2.8 Store Security Information Use Case	2-3
2.2.9 Assign User Priority Use Case	2-3
2.2.10 Assign User Privilege Use Case.....	2-3
2.2.11 Authenticate User Use Case	2-3
2.3 Interface definitions	2-4
2.3.1 Security Access Interface	2-4
2.3.2 Object Access	2-8
2.3.3 Secure Communications.....	2-8
2.4 Operation.....	2-10
2.4.1 Login Sequence	2-10
2.5 IDL	2-11

BACKGROUND

Center-to-Center (C2C) communications, like any industry wide-area network, typically address secure communications within a security policy. This policy defines the rules by which all users of the network must abide. The depth of this policy depends on the network scope as well as network risk assessment. Analysis of the risks and how much exposure is tolerable will effect costs of the communication hardware and software. This specification considers the overall effects of a security policy but does not establish the policy itself; rather, the mechanism that underlies a policy. This C2C security mechanism considers the following three security issues:

- Risk Assessment
- Vulnerability
- Security Guidelines

Risk assessment is the process of finding out what data is transmitted and how important it is. In addition to the importance of the data is the amount of damage will incur if it is lost or compromised. Traffic information is not usually considered high risk data, since it usually deals with public telemetry data from field devices and not sensitive information. Compromise of this data can disrupt but will not severely impact traffic operations and is easily recovered. On the other hand, compromise of field device control, such as changeable signs, has larger security risks because of safety and legal considerations.

Looking around the network is the first step in assessing vulnerability. Most network intrusions are from the inside, not the outside. Hackers and crackers make up just a small portion of the network compromises that are recorded annually. The vulnerability of C2C is dependent on the vulnerability of each agency connected. It is recommended therefore that each agency configure and maintain a firewall to the C2C network. It is also recommended that each agency maintain their workstations and servers in a secure area, which is locked and has restricted access. If someone can gain physical access to the servers, the servers can be compromised and also the network.

Agencies that are connected to the Internet should protect themselves and the C2C network through the proper use of firewalls. Also restricted use of dial-in modems is essential. If someone is using a modem to connect to the Internet while also connected to an agency's network, then there is a significant security hole. The best firewall is of no avail if people are bypassing them by using a modem to connect to their personal Internet accounts.

The C2C Security Service provides password protection to traffic data and device control at each center. Remote users must provide a user name and password in order to connect to the information/control interfaces. Associated with the user is his privileges and priority to access data and control devices. Passwords can be an open door to the network when poorly administered; often they are easily cracked or worse, just left laying around. A good password policy should be in place and enforced.

Section 1 GENERAL

1.1 SCOPE

This specification is written for CORBA based systems as defined by the CORBA Center-to-Center profiles. The requirements herein are tailored to the capabilities inherent in a CORBA network and will not necessarily work with other Center-to-Center protocols. The overarching objective is to give each transportation center full control over its data resources. Therefore, this specification defines a Security Service that will authenticate external users and allow them certain privileges and priorities to access data and control devices. While there are many solutions to security, this specification chooses a single approach that can be implemented using most off-the-shelf CORBA products. This specification does not impose security requirements on all transportation centers, but rather on those centers that choose secure CORBA communications to protect their resources.

1.2 PROTOCOL – LAYER RELATIONSHIP

This specification describes the required interfaces of the Security Service. Within the scheme of the ISO OSI 7-Layer Reference Model, the Security Service is within the Application and Session Layer. As a high-level service, it requires other application-layer services, such as a CORBA ORB and its associated protocols, to be in place. Table 1 shows the place of Security Service.

Table 1 - Security Service - Base Standard Relationship

NTCIP Profiles	ISO Layers	Base Standard
APPLICATION PROFILE	APPLICATION LAYER	Security Service (this standard)
	PRESENTATION LAYER	(not addressed by this standard)
	SESSION LAYER	Security Service (this standard)

1.3 REFERENCES

For approved revisions, contact:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 N.17th Street, Suite 1847
Rosslyn, Virginia 22209-3801
fax: (703) 841-3331
e-mail: ntcip@nema.org

For draft revisions of this document, which are under discussion by the relevant NTCIP Working Group, and recommended revisions of the NTCIP Joint Committee, visit the World Wide Web at <http://www.ntcip.org>.

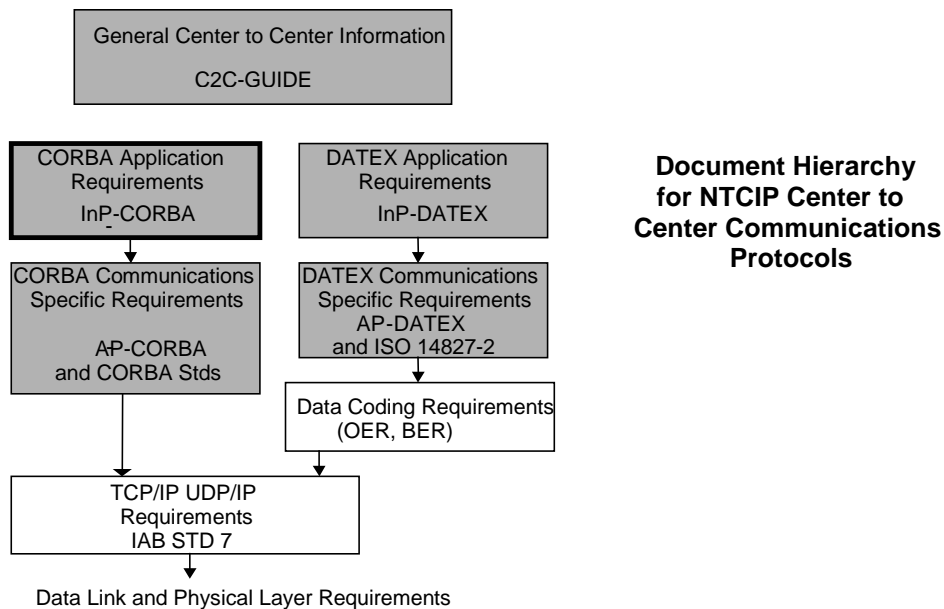
The following standards (normative references) contain provisions, which through reference in this text, constitute provisions of this Standard. Other documents and standards (other references) are referenced in these documents, which might provide a complete understanding of the structure and use of profiles. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

This document is part of a hierarchy of documents that describe system requirements for center-to-center communications. Related documents include:

- NTCIP 1104 CORBA Naming Convention
- NTCIP 1105 CORBA Near-Real Time Data Service
- NTCIP 1106 CORBA Security Service
- NTCIP 2304 Application Profile – Data Exchange ASN.1
- NTCIP 2305 Application Profile – CORBA
- NTCIP 2501 Center-to-Center Information Profile-DATEX
- NTCIP 2501 Center-to-Center Information Profile-CORBA
- NTCIP 9001 NTCIP Guide

The NTCIP Guide (NTCIP 9001) is a particularly useful reference for general information relating to the needs of inter-operating centers, the evolution of the approaches contained in the other documents, and guidance for connecting different center types for the interchange of information and control. It also explains the difference between the two center-to-center protocols supported by the NTCIP – CORBA and DATEX-ASN. The CORBA Security Service Specification applies only to centers using the CORBA standard.

The CORBA Security Service is not required for C2C compliance, but it is required for centers that wish to implement object security through CORBA. Centers that plan to use the Security Service must also support the base CORBA standard.



**Document Hierarchy
for NTCIP Center to
Center Communications
Protocols**

**Figure 1
Document Hierarchy for NTCIP Center-to-Center Communications Protocol**

1.3.1 Normative References.

The following documents and base standards provide a more complete understanding of this application profile as described herein.

The Common Object Request Broker: Architecture and Specification, Revision 2.1, The Object Management Group, Inc., July 1995, last revised September 1997. (CORBA 2.1).

Common Facilities Architecture, Object Management Group, Framingham, Massachusetts, November 1995, last revised June 15, 1997.

Common Object Services Specifications, Object Management Group, John Wiley & Sons, 1995, last revised December 2, 1997.

1.3.2 Informative References

NTCIP 8003, formerly referenced as AASHTO/ITE/NEMA TS 3.PRO, National Transportation Communications for ITS Protocol (NTCIP) Framework and Classification of Profiles.

ISO 7498:1984, Information Processing Systems--Open Systems Interconnection--Basic Reference Model.

NEMA NS 1-1995, Guide for Preparation of NEMA Standards Publications.

At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standard listed below.

AP-CORBA, current revision, National Electrical Manufacturers Association, Rosslyn Virginia.

InP-CORBA, current revision, National Electrical Manufacturers Association, Rosslyn Virginia.

AP-DATEX, current revision, National Electrical Manufacturers Association, Rosslyn Virginia.

This standard intends to be used in conjunction with AP-CORBA.

1.4 DEFINITIONS, ACRONYMS, AND TERMS

AP-CORBA	An application protocol describing center to center communication using CORBA
AP-DATEX	An application protocol describing center to center communication using Datex and ASN.1.
APL	Application Programming Language.
ASN.1	Abstract Syntax Notation One. A formal syntax-definition language used to define communications messages and data objects
ATP	Audio Transfer Protocol. A developing specification for transmission of encoded audio information, such as speech and music via the internet.
Authentication	A process used to verify the integrity, both as to originator and contents, of transmitted data, especially regarding "messages."
Center Developers	An arbitrary term referring to those representatives of governmental agencies or commercial systems developers who are implementing the hardware, software and communications infrastructure at centers. It is noted that center developers may be from the same organization as center management.

Center Management	An arbitrary term referring to those representatives of governmental agencies or commercial establishing developing new centers for transportation management.
Client	An object or program which accesses a server through one of its interfaces.
Communication Network	A collection of interconnected equipment that provides a data communications service for devices, often computers attached to the equipment.
CORBA	Common Object Request Broker Architecture.
COSS	CORBA Object Service Specifications. OMG Specifications for Services.
CS	Collection Service. CORBA 3 service, which permits the definition and use of arbitrary aggregates of data, called "Collections," all of which can be manipulated and transferred among nodes as a group.
Data Link Layer	The level of protocol as defined by ISO that provides service to transfer data between network layer entities, usually in adjacent nodes. The data link layer provides error detection (and may provide error correction) for errors occurring in the physical layer.
DATEX	Data Exchanges. Refers to the Transport Information and Control Systems – Data interfaces between centres for transport and information control systems ISO Standard 14827n.
DCE	Open Software Foundation Distributed Computing Environment.
Domain	A realm or range of knowledge, often characterized by uniquely defined terms or acronyms.
FTP	File Transfer Protocol. A standard protocol to transfer files from computer to computer on the Internet.
IAB	Internet Architecture Board, an organization which is responsible for the development and publication of Internet protocols.
IDL	Interface Definition Language. A descriptive language used in CORBA to specify the interface through which clients may access servers to obtain access to objects.
IEC	The International Electrotechnical Commission, an international standards organization similar to the ISO, but having both member nation standards bodies and selected commercial representation. By agreement with the ISO, the IEC is primarily involved with standard relating to the hardware of communications while the ISO is primarily involved with standards relating to the software of communications.
IIOIP	Internet Interoperability Protocol
Interface Repository (IR)	A public, on-line storage of the IDL interface definitions applicable to one (or more) CORBA-compliant centers. This store is supported by a standard CORBA Service for storing IDL definitions. Most especially, a national repository, connected to the internet, containing in a database, descriptions

for objects and their attributes, including that data attributed to data dictionaries and message catalogs. See Registry.

IOR	Interoperable Object Reference. Object reference protocol associated with IOP.
ISO	International Organization for Standardization, a voluntary organization consisting of designated standards bodies of participating world nations. By agreement with the IEC, the ISO is primarily involved with standard relating to the software of communications while the IEC is primarily involved with standards relating to the hardware of communications.
ITS	Intelligent Transportation Systems.
Layer	A group of services, and functions that is conceptually complete, that is one of a set of hierarchical levels forming a complete domain, and that extends to all system nodes which are compliant with the network architecture.
Level	See Layer. Most commonly used in European Communications Standards
LRMS	Location Reference Message Specification. Federal Standard for location by Oakridge Laboratories
Network	A collection of nodes, domains, clients, servers, communications channels interfaces and applications connected together physically and logically
Network Layer	OSI Model Level 3. Carries out routing of data through a communications network.
Node	Abstraction over the notion of computer.
NRTDS	Near-Real-Time-Data Services. An additional specification for the use of CORBA within NTCIP Center-to-Center Compliant Systems to provide distribution of frequently changing data (down to a granularity of about 1 second)
NS	Naming Service. Standard CORBA Service for location of objects by name.
NTCIP	National Transportation Communications for ITS Protocol. An initiative to provide a communications standard that ensures the interoperability and interchangeability of traffic control and Intelligent Transportation Systems (ITS)
Object Reference	Instance of an interface, defined in IDL. A client holding it may use it to access the server which provides the implementation object to obtain access to its attributes.
OMG	Object Management Group. Industrial consortium working to standardize open distributed object-based computing.
ORB	Object Request Broker.
OSF	Open Software Foundation, an organization of major computer manufacturers and software vendors which support the development of standards for interoperability of computer hardware, communications and software systems.

Physical Layer	ISO model Layer 1. Covers the electrical, mechanical and timing aspects of signal transmission over a physical medium of communication.
Registry	An on-line storage facility for information describing a center, the protocols which it follows, the data objects and attributes which it supports, and the information (messages, function calls, etc.) which it will exchange with other centers. The notion of registry in NTCIP includes, but is broader than, the concept of Interface Repository (IR) in the CORBA specification.
Service	In communications, a series of messages defined between applications level protocols, which, in the aggregate, carry out a single activity understandable to a user of the communications system.
SS	Security Service. Standard CORBA service for verifying that requestors of objects have permission to access the objects
TCP/IP	Transmission Control Protocol. A transport level protocol used on the Internet to provide connection-based communication between pairs of communicating TCP user processes. The protocol does not guarantee the delivery of data but does prevent duplication of messages, and maintains order on the data presented to the receiver. TCP supports the signaling of control data (or "urgent" data) within the data stream.
TFTP	Trivial File Transfer Protocol.
TMC	Transportation Management Center, also called Center
TS	Trader Service. Standard CORBA service for locating objects by attributes
UDP/IP	User Datagram Protocol. A transport level protocol used on the Internet to provide connectionless data exchange for applications level procedures. The protocol does not guarantee the delivery of data nor prevent duplication of messages. The protocol is, however, low in overhead, and may be highly useful in the transmission of data which is inherently lossy (such as audio and video frames) or which spoils if not delivered in a timely fashion.
WG	Working Group. The NTCIP Center-to-Center working group tasked with developing this standard (or other standards) as a part of the overall NTCIP initiative

Section 2

REQUIREMENTS AND CONFORMANCE

2.1 GENERAL REQUIREMENTS

This section describes the functions necessary for the Security Service to fulfill the CORBA security task. The Security Service consists of the following functions:

- User authentication
- Secure communications
- Access privilege
- Access priority
- Access reporting

The interface roles that apply to this specification are:

- User login to remote center
- Control of remote user's access to center's information and device control

2.2 FUNCTIONAL REQUIREMENTS

Implementations claiming conformance to this CORBA Security Specification shall support the following functional requirements as described and shown in use cases of Figure 2. When a user accesses a center's data, he first logs into that center's Security Service (SS). Upon authentication by the SS, the user is assigned an access level or priority linked to the center's devices, events, and services. The user remains logged in until he explicitly logs off through the SS. At any time, the user can check on his login status including his access level and priority by requesting that information from the SS.

The SS maintains the security access on a per user basis. Each user is assigned privileges and priority by the system administrator. Access assignments are per device and method/attribute privileges are per device type. Access assignments for services (e.g. location translation) are per service and method/attribute privileges are per service type. Because of their transitory nature, event assignments are on a per event type method/attribute only.

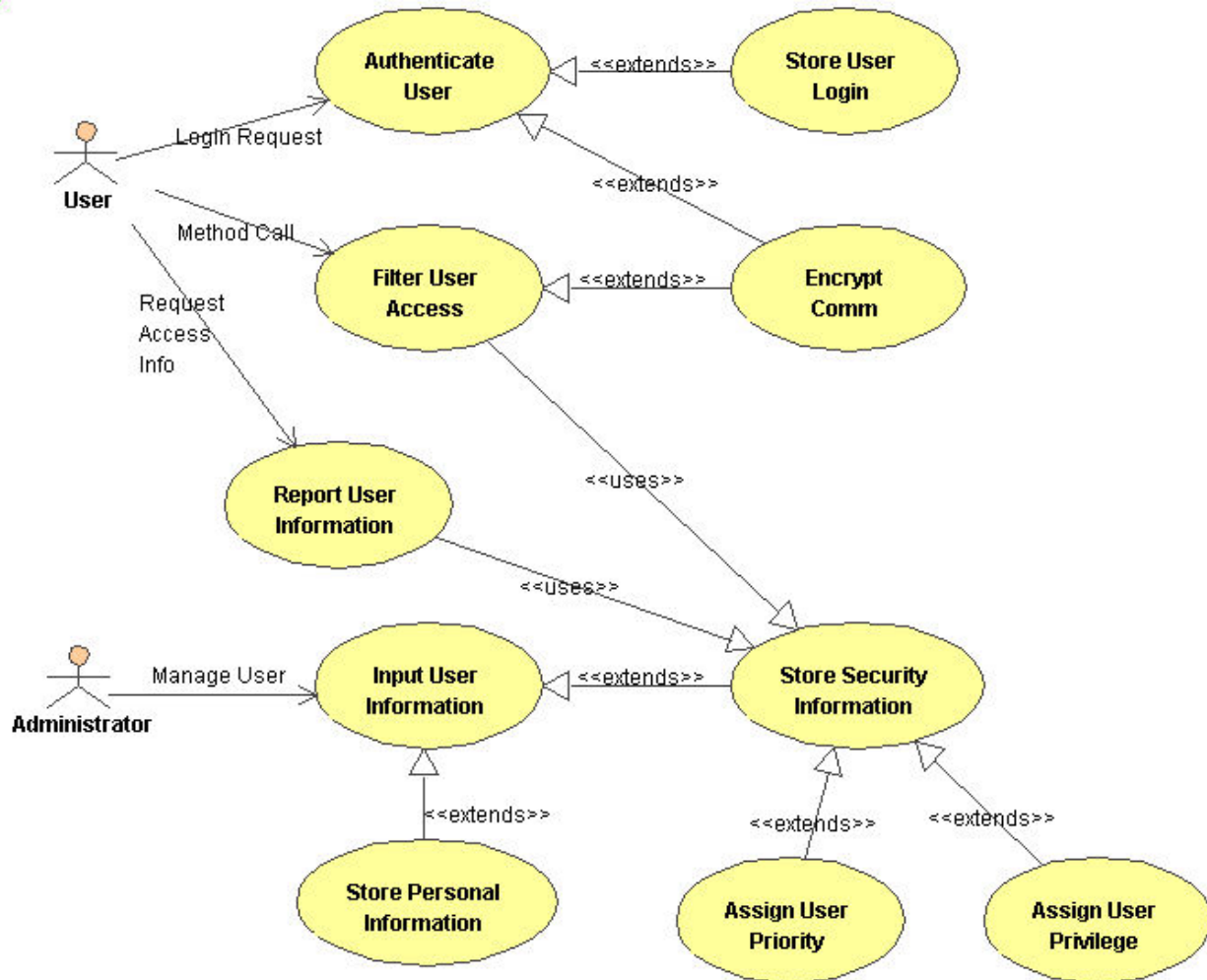


Figure 2
CORBA Security Use Case

2.2.1 Administrator Actor

The administrator actor is a local client to the Security Service. His purpose is to manage the remote user database and security tokens. From the administration function, user name, password, priority, and privilege is maintained.

2.2.2 User Actor

The user actor is any remote application that acts as a client to secure servers. Remote clients will login to the Security Service using secure communications, then after a successful login, will call methods on the servers of interest.

2.2.3 Filter User Access Use Case

Filtering of object access shall control users access to object's data and control by allowing or denying method calls. Each method call invoked upon a server shall be intercepted by the security filter and checked against the security tokens active at that server. Object attribute, object method, and the object itself shall be filtered based upon the Access Control List for that user. If access is allowed then the server shall execute the method call; if denied, the server shall raise an exception of access denied.

2.2.4 Encrypt Communication Use Case

Secure communications shall allow the transmission of user name, password, and data without tampering or unauthorized discovery through eavesdropping. The transmission of this data shall be by Secure Socket Layer (SSL) protocol.

2.2.5 Report User Information

The user can at any time request from the Security Service (SS) information on his login status. The SS shall supply user login status including access privilege and priority when requested by an authorized user.

2.2.6 Store User Password Use Case

The Security Service shall store and maintain a list of passwords. Each password stored in the system shall be protected from unauthorized viewing using encryption. Encryption shall use a proven algorithm, such as DES. The SS shall accept passwords that contain a mix of letters, numbers, and special characters.

2.2.7 Input User Information Use Case

The security administrator manages external users through the security service running in his local center. The system shall allow the administrator to add, modify, or delete external users. The administrator can set up new user accounts by entering user information and Access Control Lists information.

2.2.8 Store Security Information Use Case

The Security Service shall store and maintain Access Control Lists (ACL) for each user. The ACL consists of devices, events, and services that are allowed a given user at a given center. The ACL defines the privileges for a user with a list of devices, and method calls for a given device type. For example, an ACL for a user's CCTV privilege will contain the list of cameras (e.g. camera 1, 2, and 5) and camera methods (e.g. pan, tilt, and zoom).

2.2.9 Assign User Priority Use Case

Priority logic defines a user's ability to override another user of lower priority. The Security Service shall assign user priorities from 0 to 9, where 0 is the highest priority user. The Security Server shall allow users to break device locks and take control of that device from a lower priority user. Lower priority users shall not be able to break the lock of a higher priority user.

2.2.10 Assign User Privilege Use Case

Privilege logic defines a user's rights to access objects and to control them. Privilege shall allow access to objects based upon the ACL for a given user.

2.2.11 Authenticate User Use Case

User authentication shall require all users to login with user name and password. Name and password is compared against stored information to validate user. Name and password shall match exactly without difference in case before the user is considered a valid user by the system. The user shall retain the status of valid user until logout by user or system.

2.3 INTERFACE DEFINITIONS

The typical security system consists of servers and clients as shown in Figure 3. Within the security system, the Security Service has interfaces with the following other systems:

- Remote applications that logon or logoff from another center
- Remote applications that need access information about a remote user
- Remote applications that call methods on local objects

Security Service communicates with the above systems via the following interfaces:

- Security Service – Secure Access
- Security Service – Object Access

Secure Access is the interface to the Security Server, as shown. Object Access is the interface to any CORBA object that is providing transportation functions or data. This interface can be secure or non-secure. Secure Access and Object Access impose the following requirements on their respective interfaces:

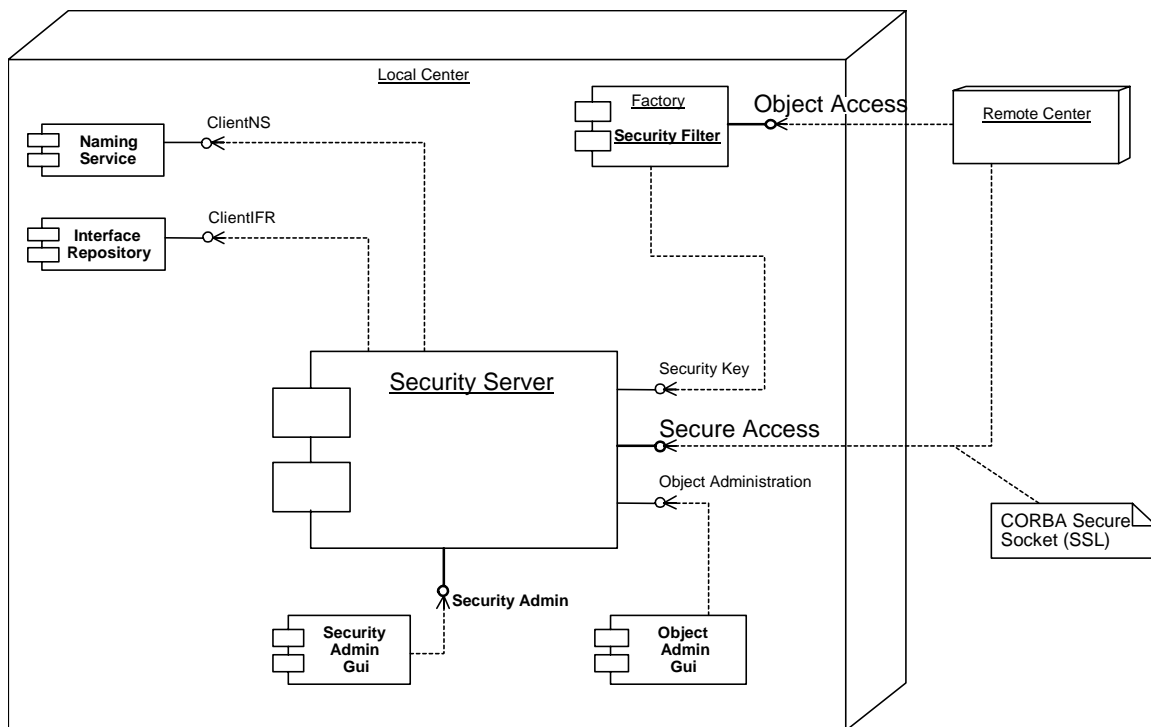


Figure 3
Security System

2.3.1 Security Access Interface

Secure Access shall have the following CORBA interfaces and associated methods. This interface is an implementation of user authentication, and access reporting from a local center to a remote center. User authentication implements a center login with user name and password, while access reporting implements a report of user privileges for requesting user. The Secure Access class diagram shown in Figure 4 defines the methods and attributes for this interface. This interface is the primary interface into the

Security Service and depicted in Figure 4. It supplies the methods for user login, user logout, and user access reporting. Secure information (Secure Info) is detailed in Figure 5.

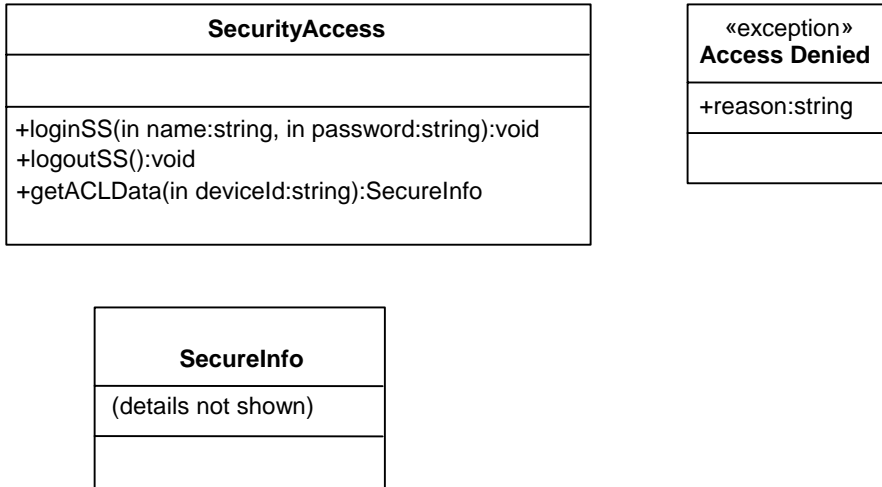


Figure 4
Security Access Class Diagram

getACLData Method

getACLData(deviceId:string):SecureInfo

This method allows the authorized client to receive token data based upon the standard device type. For example, a client can request and receive data on which cameras and which camera methods he has access privilege. The login process must be completed before calling this method. If successful, token data of device type is returned. If not successful, a null value is returned.

loginSS Method

loginSS(name:string, password:string):void

This method allows the client to pass in his or her name and password. The security service checks this information against the password file and returns Access Denied if no match or nothing if matched. If login is successful then security tokens are sent to the local servers to allow this client.

logoutSS Method

logoutSS():void

This method allows the client to logout of the current center. If logout is successful then all security tokens are removed from the servers and the user must login once again to gain access. If logout fails then Access Denied is returned.

Access Denied Class

This Access Denied exception is thrown whenever users are denied access during the validation process.

Secure Info Class

This class is security information that is viewable but not modifiable by the client. It contains device and method access information. Secure Info consists of the information relevant to the user and his privileges. The class diagram of Figure 5 shows the attributes that this interface provides.

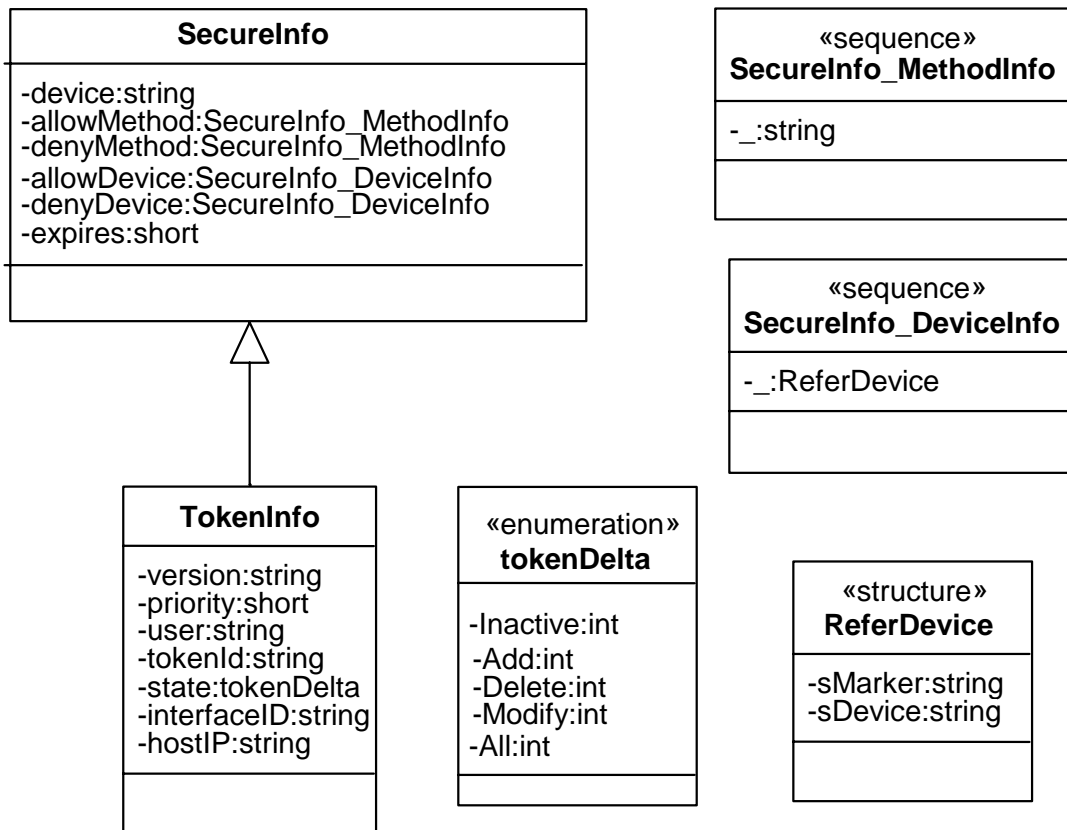


Figure 5
User Security Information (ACL)

allowMethod Attribute

allowMethod:SecureInfo_MethodInfo

This method defines which object methods the current client (CORBA principal) has access privileges. If this sequence is empty then all methods are denied unless the "denyMethod" has some methods; in this case, those in the "denyMethod" are denied the rest is allowed. If this sequence has some or all methods then the client is allowed privilege to access them.

denyMethod Attribute

denyMethod:SecureInfo_MethodInfo

This method defines which methods are not accessible by the current client (CORBA principal). If this sequence is empty then all methods are allowed unless the "allowMethod" has some or no methods. If this sequence has some or all methods then the client is denied privilege to access them.

device Attribute

device:string

Standard device type defined for the network.

allowDevice Attribute

allowDevice:SecureInfo_DeviceInfo

This method defines which objects (devices) are allowed access by the current client (CORBA principal). If this sequence is empty then all objects are denied unless the "denyDevice" has devices; in this case, those in the "denyDevice" are denied the rest is allowed. If this sequence has some or all devices then the client is allowed privilege to access them.

denyDevice Attribute

denyDevice:SecureInfo_DeviceInfo

This method defines which devices are not accessible by the current client (CORBA principal). If this sequence is empty then all objects are allowed unless the "allowDevice" has some or no devices. If this sequence has some or all devices then the client is denied privilege to access them.

expires Attribute

expires:short

This attribute indicates the number of days before this token expires and is no longer valid. Once expired, this token is revoked and all servers deny the client access. Units are the number of days until expiration. Usually, this number is counted down by the security service or by the servers themselves.

SecureInfo_DeviceInfo Class

This sequence contains the device names and markers for identifying access to these devices. Markers reference the device object.

SecureInfo_MethodInfo Class

This sequence contains the method names extracted from the Interface Repository for a given device. Names are used to identify access to the device methods. The Interface Repository contains IDL for any given interface.

tokenDelta Class

Enumerated values for the state of Tokens. Where "add" means this token is new, "delete" this token is to be removed, "modified" this token has changed information.

TokenInfo Class

This is the secret part of the Token that is passed to the client process as part of any type object. The client does not see this part of the IDL, which changes the tamper algorithm periodically.

hostIP Attribute

hostIP:string

The host id attribute contains the IP address of the current registered user. The address format is the fully qualified host name string.

interfaceID Attribute

interfaceID:string

The attribute contains the Interface Repository (IFR) identification for the given device/event interface, as stored in the IFR.

priority Attribute

priority:short

Priority of the principal usually based upon whether the user is external or internal. Priority range is from 0 to 9, where 0 is the highest priority.

state Attribute

state:tokenDelta

This attribute contains the current state of a token whether it is added, deleted, or modified.

tokenId Attribute

tokenId:string

This attribute contains a unique identification for the token.

user Attribute

user:string

This attribute is the CORBA principal that will make the object call.

version Attribute

version:string

Version identifier used to check against the versions in the IFR.

2.3.2 Object Access

The Object Access pertains to any CORBA interface that has external access permitted. Object Access works in conjunction with Secure Access to filter method calls made by remote users on local servers. If authorized, the access succeeds otherwise an exception is returned.

2.3.3 Secure Communications

The Security Service shall use Secure Socket Layer (SSL) security protocol when secure communications is needed between centers. SSL is layered between the transport-level protocol, TCP/IP, and the inter-ORB protocol, IIOP (shown in Figure 6). SSL provides server authentication, privacy, and integrity for communications across TCP/IP. Server authentication allows a client application to verify the identity of server with which it communicates. Privacy ensures that transmitted data between client / server cannot be eavesdropped on or understood by a third party. Integrity allows the client / server transactions to detect if data was modified during transmission.

SSL Authentication

SSL uses Rivest Shamir Adleman (RSA) public key cryptography for authentication. Public key cryptography has an associated public and private key, where data encrypted with the private key can be decrypted only with the public key. Public key cryptography allows a secure server to prove its identity by encoding data with its private key. Any client application can check the content of the encoded data by decoding it with the server's public key.

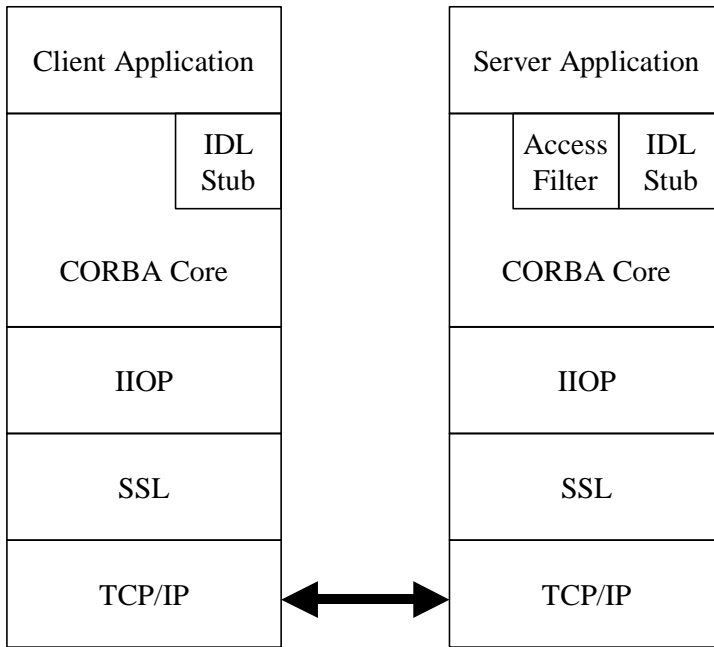


Figure 6
Communication Stack with SSL

With SSL, a secure server transfers the public key to the client using a certificate from a trusted certificate authority (CA). SSL authentication shall use X.509 certificates to transfer information about a client's public key.

Once a client receives a public key from the server, all subsequent communications between the two applications can be encoded using an encryption algorithm. The SSL algorithm shall be the Data Encryption Standard (DES).

Communication Invocation Policy

A fully secure C2C interface uses SSL in all of its servers. The least secure system is one in which no servers uses SSL security. It is envisioned that a combination of SSL protected servers and non-SSL protected servers will make up the majority of systems; this is due to the nature of ITS data and control. C2C shall allow systems to set their own communication invocation policy. Servers can be setup as always secure, always non-secure, or accept both secure and non-secure connections.

Security Guidelines

The C2C security policy defines the guidelines by which all users of the network must abide. These guidelines can be defined in the following categories:

Acceptable use - The kind of activity deemed acceptable and unacceptable on the C2C network.

Access - Who has access to the network and of those people, the areas they are granted access and areas they are denied access.

Privacy - What kinds of security monitoring will be going on? (i.e. monitoring of email, server access, logging of individual users through electronic means, etc.).

Passwords - Guidelines for user passwords such as minimum length, types of characters used or even if they will be assigned rather than chosen by the user.

Enforcement - How to report user violations of policy and what actions will be taken for violations.

Purchasing - How to regulate the purchase and implementation of new hardware and software.

Support and Maintenance - Who is responsible for the upkeep of the C2C network and servers, and who will maintain security in the event of changes or alterations to the system.

2.4 OPERATION

The primary sequence for the Security Service is the login sequence. Users shall login once to each center and remain logged in until an explicit logoff by user or center.

2.4.1 Login Sequence

The Security Service login sequence to a single center consists of the steps that a remote user (client process) would take to establish a connection to any secure object.

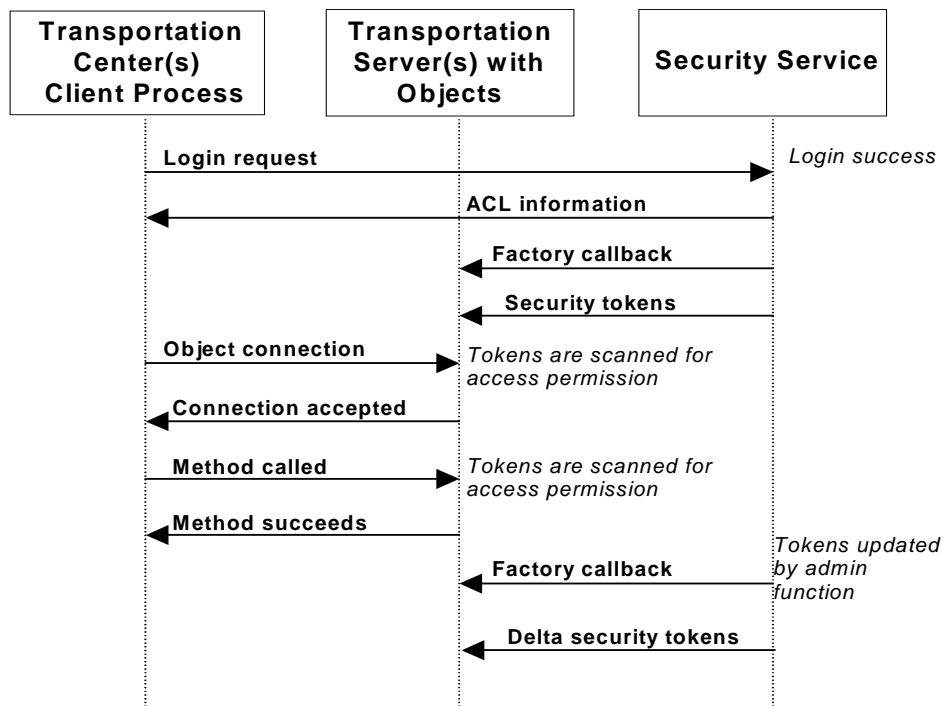


Figure 7
Security Service Sequence

Security Service (SS) sequence is as follows:

1. A client process from a remote Transportation Center sends a login request to SS.
2. SS authenticates the user and sends access control list (ACL) information back.
3. Simultaneously, the SS sends a callback to all registered factories.

4. The factories request and receive the client's security tokens for their object type.
5. The client process attempts an object connection.
6. The factory's filter authenticates the connection and accepts it.
7. The client process calls a method to control or receive data.
8. The factory's filter verifies access rights and passes the request to the object.
9. The object executes the request, performing control or passing back data.
10. The administrator updates token data.
11. The SS sends a callback to all registered factories
12. The factories request and receive the token updates.

2.5 IDL

```
/*
 *
 * PROJECT      : security
 * FILE NAME    : security_service.idl
 * BASELINE     : /project/cm/security/baseline
 * DEVELOPERS   : mosl
 * DATE TIME    : Mon Jul 28 17:02:00 1999
 *
 * REMARKS:
 *
 */

#ifndef SECURITYSERVICE_IDL
#define SECURITYSERVICE_IDL

module SecurityService {

    interface TokenInfo;

    // Enumerated values for the state of Tokens.  Where "add" means
    // this token is new, "delete" this token is to be removed,
    // "modified" this token has changed information.  "All" means
    // that Add or Modify states have been around for more than
    // one update cycle and move out of the delta queue to the all queue.
    //
    enum TokenDelta { Inactive, Delete, Add, Modify, All };

    // Exception thrown whenever user access is denied by the
    // security validation process.
    //
    exception ACCESS_DENIED { string reason; };

    // Device ID / Marker pair, where marker indicates the object
    // reference.
    struct ReferDevice
    {
        string sMarker;
        string sDevice;
    };
};
#endif
```

```
// This sequence contains the method names extracted from the
// Interface Repository for a given device. Names are used to
// identify access to the device methods.
//
typedef sequence<string> SecureInfo_MethodInfo;

// This sequence contains the device names and markers for
// identifying access to these devices. Markers reference
// the device object.
//
typedef sequence<ReferDevice> SecureInfo_DeviceInfo;

// This structure contains the secure user information accessible
// by the Security Admin functions. Data is stored for each user
// to allow access to objects.
//
struct UserInfo
{
    string loginName;
    string password;
    string email;
    string firstName;
    string lastName;
    string agency;
    short priority;
    string userStatus;
};

struct TokenData
{
    string token_device;
    SecureInfo_MethodInfo token_allowMethod;
    SecureInfo_MethodInfo token_denyMethod;
    SecureInfo_DeviceInfo token_allowDevice;
    SecureInfo_DeviceInfo token_denyDevice;
    short token_expires;
    //
    string token_version;
    short token_priority;
    string token_user;
    string token_tokenId;
    TokenDelta token_state;
    string token_interfaceID;
    string token_hostIP;
};

// This sequence contains the Tokens assigned to a user when
// registered with the security service.
//
typedef sequence<TokenData> SecureInfo_MyTokens;

// These tokens are associated with a specific device or service
// type.
```

```
//
typedef SecureInfo_MyTokens DeviceTokens;

// This structure contains an array of security tokens assigned
// to a given user. The tokens are the information which allow
// or deny access to devices and device methods.
//
typedef SecureInfo_MyTokens UserTokens;

// This interface is the security information that is viewable
// but not modifiable by the client. It contains device and
// method access information. Tamper check is done to ensure no
// tampering with these values.
//
interface SecureInfo
{
    // Standard device type, defined for the network.
    //
    readonly attribute string device;

    // This method defines which object methods are allowed by the
    // current principal.
    //
    readonly attribute SecureInfo_MethodInfo allowMethod;

    // This method defines which methods are not accessible by the
    // current principal.
    //
    readonly attribute SecureInfo_MethodInfo denyMethod;

    // This method defines which object (devices) are allowed by the
    // current principal.
    //
    readonly attribute SecureInfo_DeviceInfo allowDevice;

    // This method defines which devices are not accessible by the
    // current principal.
    //
    readonly attribute SecureInfo_DeviceInfo denyDevice;

    // This attribute indicates when this token expires and is no
    // longer valid. Units are in days until expired.
    //
    readonly attribute short expires;
};

// This interface is the primary interface into the Security
// Service. This is where the client authentication / login
// is preformed and the user security token is given out to
// the factories.
//
interface SecurityAccess
{
    // This method allows the client to pass in his or her name
```

```
// and password. The security service checks this information
// against the password file and returns a true is successful
// or a false is no match.
//
void loginSS(in string name, in string password)
raises (ACCESS_DENIED);

void logoutSS()
raises (ACCESS_DENIED);

// This method allows the authorized client to receive a token
// based upon the standard device type. The login process must
// be completed before calling this method. If successful, a
// token of device type is returned. If not successful, a null
// value is returned.
//
SecureInfo getACL(in string deviceId)
raises (ACCESS_DENIED);

TokenData getACLData(in string deviceId)
raises (ACCESS_DENIED);
};

// This interface allows the administrator to add, delete, and
// modify users to the system. This is a secure interface
// which provides setting and viewing user and security
// information. Tokens are retrieved by device name or by
// specifying "all" to get all tokens for that user. Tokens are
// set by specifying a single device or "all" which removes all
// existing tokens and adds the newTokens. Remove tokens can be
// "all" or specific device.
//
interface SecurityAdmin
{
// Association accessor methods
//
UserInfo getUser(in string userName);
void setUser(in string userName, in UserInfo newUser);
void removeUser(in string userName);

UserTokens getTokens(in string userName);
void setTokens(in string userName, in UserTokens newTokens);
void removeTokens(in string userName);
};

// This interface allow factories to receive tokens for client
// access to their objects. Factories and receive a full set of
// active tokens or only those tokens that have changed.
//
interface SecurityKey
{
void cbSignOn(in string cbIOR, in string tokenType);
void cbSignOff(in string cbIOR);

DeviceTokens getDelta(in string device);
DeviceTokens getAll(in string device);
};
```

```
};

// This is the secret part of the Token, which is passed
// to the client process as part of an any type object.
// The client does not see this part of the IDL.
//
interface TokenInfo : SecureInfo
{
    // Version identifier used to check against the versions
    // in the IFR.
    //
    readonly attribute string version;

    // Priority of the principal with regard to this token's
    // device type.
    //
    readonly attribute short priority;

    // The principal that is making the object call.
    //
    readonly attribute string user;

    // This attribute contains a unique identification for the token.
    //
    readonly attribute string tokenId;

    // This attribute contains the current state of a token whether
    // it is added, deleted, or modified.
    //
    readonly attribute TokenDelta state;

    // The descriptive text for the idl from the interface repository
    //
    readonly attribute string interfaceID;

    // The ip address of the user
    //
    readonly attribute string hostIP;
};

};

#endif /* SECURITYSERVICE_IDL */
```