CHAPTER 4

Secure Socket Layer Support

Revised: July 2010, OL-23043-01

This chapter describes the Common Object Request Broker Architecture (CORBA) Secure Socket Layer (SSL) support.

The BTS 10200 provides a secure CORBA transport using an SSL module in the CORBA adapter CORBA interface servant (CIS). The Object Management Group (OMG) organization defines the Common Secure Interoperability Specification, and Version 2 (CSIv2) defines the Security Attribute Service (SAS) protocol. The SAS enables interoperable authentication, delegation, and privileges.

The SAS protocol exchanges its protocol elements in the service context of the General Inter-ORB Protocol (GIOP) request and reply messages that are communicated over a connection-based transport. The protocol is intended to be used in environments where transport layer security, such as that available using SSL/transport layer security (TLS) or Internet Inter-ORB Protocol (IIOP) over SSL (SSLIOP), provides message protection (that is, integrity and or confidentiality) and server-to-client authentication. The protocol provides client authentication, delegation, and privilege functionality that can be applied to overcome corresponding deficiencies in an underlying transport. The SAS protocol facilitates interoperability by serving as the higher-level protocol under which secure transports can be unified. The CIS implementation of SAS provides the following:

- Secure interoperability predicated on the use of a common transport-layer security mechanism, such as that provided by SSL/TLS.
- Message protection as necessary that is provided by the transport layer to protect GIOP input and output request arguments.
- Target-to-client authentication is provided by the transport layer to identify the target.
- Transport-layer security ensures that the client does not have to issue a preliminary request to establish a confidential association with the intended target.
- Support for clients that cannot perform authentication by using transport-layer security mechanisms; the SAS protocol provides client authentication above the transport layer.
- To support the formation of security contexts using GIOP service context, the SAS protocol requires at least one message in each direction to establish a security context.
- Support for security contexts that exist only for the duration of a single request/reply pair.
- Support for security contexts that can be reused for multiple request/reply pairs.

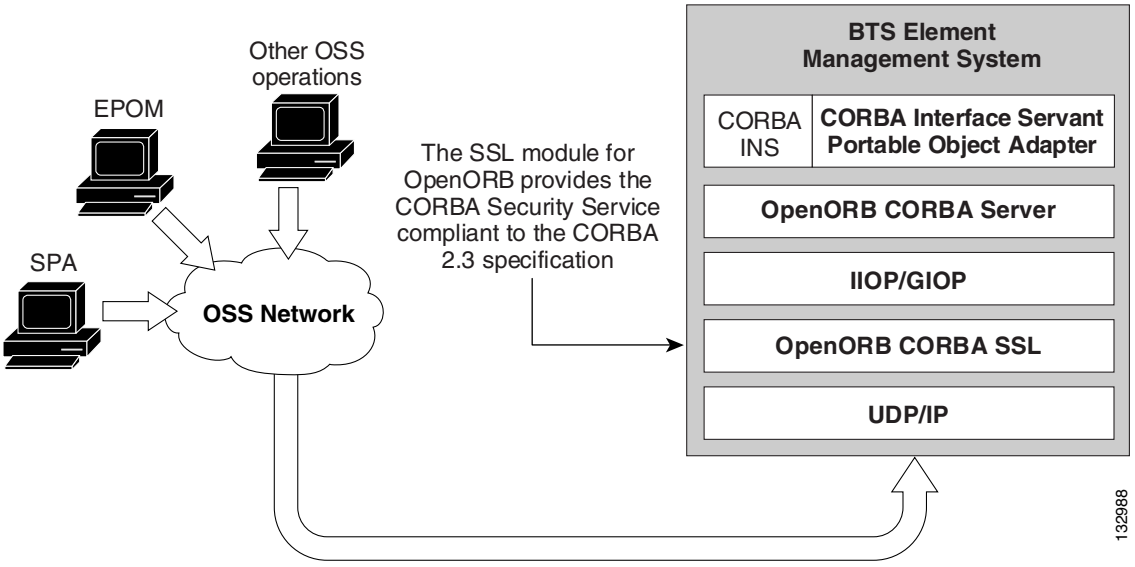
This implementation is provided through a module extension supplied in the OpenORB CORBA distribution.

System Context for System Security Extensions

This section describes the system context for system security extensions. The system security extensions consist of the SSL module for CORBA combined with the exchange of security certificates.

Figure 4-1 provides an overview of the components of the BTS 10200 that are affected by the system security extensions. The exception is the base Solaris OS platform.

Figure 4-1 CORBA Implementation of Secure Sockets Layer



Dependencies

This section defines the dependencies for the application components in the Security Extension feature. The dependencies are largely based on external components.

Reduced Solaris Image

There is no specific dependency for this application component. However, the dependency is based on the security needs of the BTS 10200.

Java Implementation of SSL

The CIS application program depends on the OpenORB implementation of the SSL component module for the CORBA Security Support. This OpenORB module in turn requires the Java implementation of the JSSE or Java Secure Sockets from Sun Microsystems.

Certificate and Key Password

This section describes the certificate and the key password for SSL CORBA. The primary feature of SSL CORBA is encrypted transport. This implementation uses a public-keyed, self-signed certificate only and all users have the same key password.

This section also describes the directory structure where the key store, trust store, and certificate reside for SSL CORBA and the naming conventions that must be adhered to for these files. In the BTS 10200, the certificate and key stores are delivered in the BTS 10200 CIS package. They are located at

/opt/BTScis/cert

The certificate and key stores are also available in the CORBA SDK package BTSxsdk located at

/opt/BTSxsdk/cert

The trust store and key store must be named as follows:

bts10200_ks bts10200_ts

The mandatory key password that is **Chillan** (this is case sensitive).

When you use the cis-install.sh script to install CORBA, *SSLIOP enabled* is the default setting. The client system cannot get in without using the aforementioned key password, key store, and certificate. The key store and trust store must be on both Element Management Systems (EMSs), because there is no automatic redundancy.

Note The following commands build the key and trust stores on the BTS 10200 and on the client side. Although they are built and deployed by default in the BTS 10200 and in the BTSxsdk, they can be rebuilt or replaced by hand. This is not a recommended procedure at this time.

Password: Chillan

Step 1 Generate keys (validity 8 years):

```
keytool -genkey -alias bts10200 -keyalg RSA -validity 2840 -keystore bts10200_ks
```

Step 2 Export a certificate:

```
keytool -export -alias bts10200 -keystore bts10200_ks -rfc -file bts10200.cer
```

Step 3 Import the certificate into the truststore:

```
keytool -import -alias bts10200 -file bts10200.cer -keystore bts10200_ts
```