

Oracle® Communications Session Border Controller TSCF SDK Guide



Release 1.5.0
F20326-01
July 2019

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2004, 2018, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

About This Guide

1 Overview

TSM Tunnel	1-2
SDK Host Operating System Relationship	1-3
Provided Functionality	1-4

2 Compiling the TSM Library and Documentation

SDK Directories	2-2
Download and Compile OpenSSL	2-2
Generate the API Documentation	2-3

3 Accessing and Using the TSM SDK APIs

Sample TSM SDK-based Applications	3-1
Using The SDK To Create A TSM Tunnel	3-2
Enabling Redundancy	3-4
Error Codes	3-4

About This Guide

The Oracle® Communications Tunneled Session Controller SDK Guide describes the client-side SDK (software development kit) that facilitates the creation of secure tunnels between a client application and the Tunneled Session Controller Function (TSCF) of the Oracle Communications Session Border Controller. A client is typically a softphone application that utilizes the SDK software libraries and source code to create TLS tunnels to a TSCF service, thus achieving secure real time communications and ubiquitous firewall traversal.

This document specifically describes the SDK, functional libraries, and source code supplied with the SDK Version 1.5.0.

Documentation Set

The following table describes the documentation set for this release.

Document Name	Document Description
Acme Packet 3900 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 3900.
Acme Packet 4600 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 4600.
Acme Packet 6100 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6100.
Acme Packet 6300 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6300.
Acme Packet 6350 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6350.
Release Notes	Contains information about the current documentation set release, including new features and management changes.
ACLI Configuration Guide	Contains information about the administration and software configuration of the Service Provider Oracle Communications Session Border Controller.
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Maintenance and Troubleshooting Guide	Contains information about Oracle Communications Session Border Controller logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
MIB Reference Guide	Contains information about Management Information Base (MIBs), Oracle Communication's enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.

Document Name	Document Description
Accounting Guide	Contains information about the Oracle Communications Session Border Controller's accounting support, including details about RADIUS and Diameter accounting.
HDR Resource Guide	Contains information about the Oracle Communications Session Border Controller's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about the Oracle Communications Session Border Controller's support for its Administrative Security license.
SBC Family Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the Oracle Communications Session Border Controller family of products.
Installation and Platform Preparation Guide	Contains information about upgrading system images and any pre-boot system provisioning.
Call Traffic Monitoring Guide	Contains information about traffic monitoring and packet traces as collected on the system. This guide also includes WebGUI configuration used for the SIP Monitor and Trace application.
HMR Resource Guide	Contains information about configuring and using Header Manipulation Rules to manage service traffic.
TSCF SDK Guide	Contains information about the client-side SDK that facilitates the creation of secure tunnels between a client application and the TSCF of the OCSBC.
REST API Guide	Contains information about the supported REST APIs and how to use the REST API interface.

Revision History

Date	Description
December 2018	<ul style="list-style-type: none"> Initial release
July 2019	<ul style="list-style-type: none"> Adds note about finding information about platform limitations in the release notes of S-CZ8.0.0 and higher.

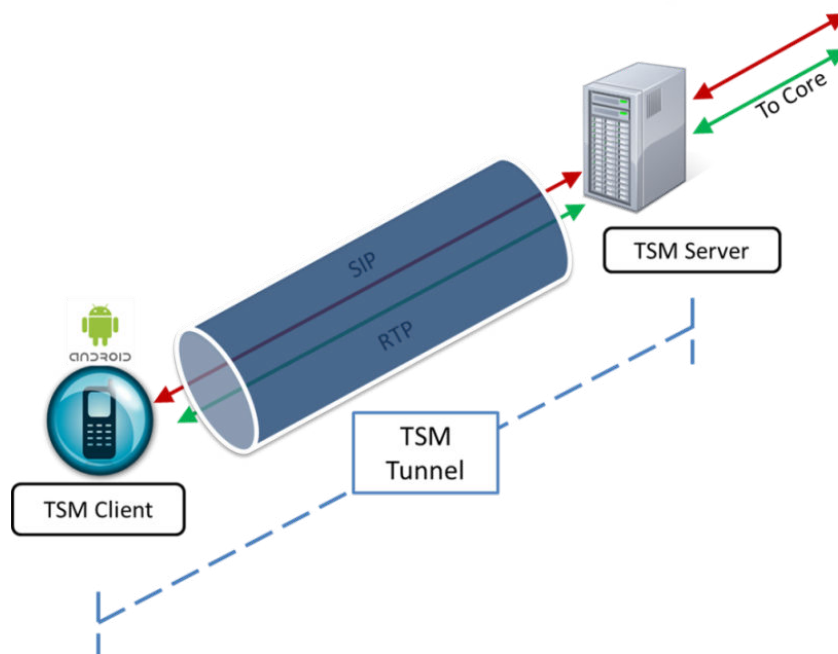
1

Overview

Tunnel Session Management (TSM) improves firewall traversal for real time communications for OTT VoIP applications and reduces the dependency on SIP/TLS and SRTP by encrypting access-side VoIP within standardized VPN tunnels. As calls or sessions traverse a TSM tunnel, the Oracle Communications Session Border Controller (OCSBC) will route all SIP and RTP traffic from within the TSM tunnel to the core (or appropriate destination).

Oracle Communications is working with other telecom providers and vendors to standardize TSM. Within the 3GPP, TSM is called a Tunneled Services Control Function (TSCF). Currently the 3GPP Technical Requirement draft is TR 33.8de V0.1.3 (2012-05) as a standardized approach for overcoming non-IMS aware firewall issues with supporting companies including China Mobile, Ericsson, Huawei, Intel, RIM, Vodafone, and ZTE. Beyond the standard, we provide exceptional tunnel performance & capacity within the OCSBC as well as high availability, DDoS protection and our patented TSM Tunnel Redundancy to improve audio quality in lossy networks such as the Internet.

Figure 1-1 Basic TSM Setup



TSM consists of two parts:

- the TSM server (often referred to as a TSCF or Tunneled Services Control Function)
- the TSM client

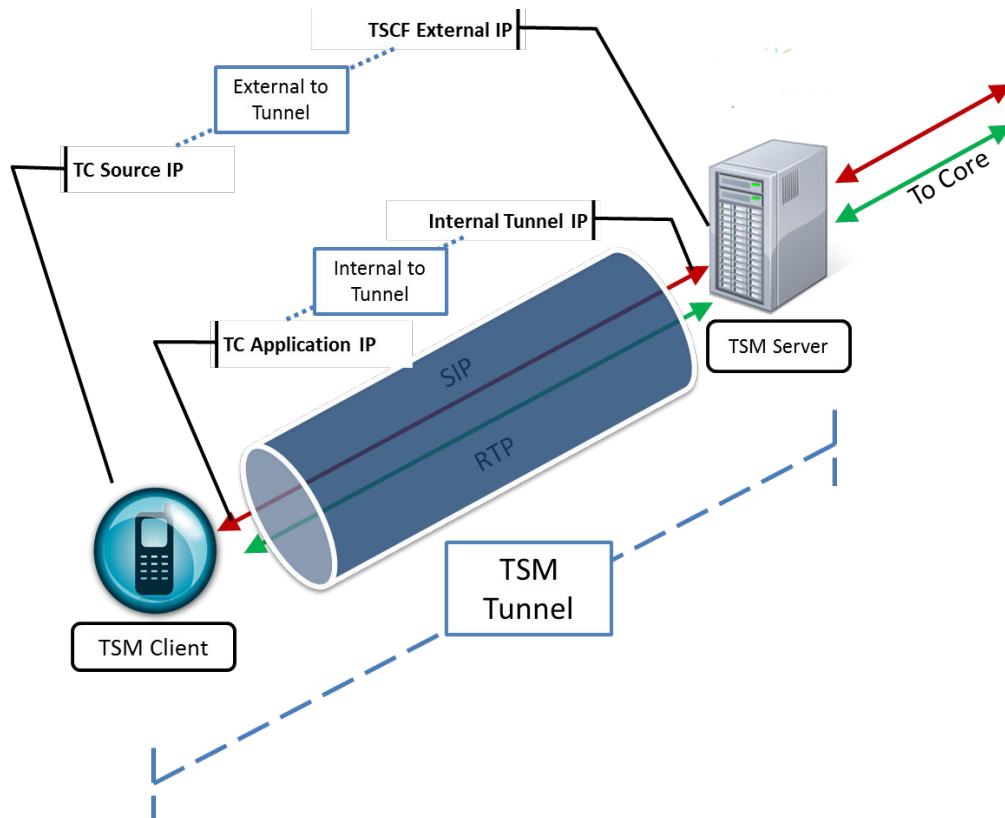
The TSM server resides and runs on the OCSBC and the TSM client runs within applications that reside on workstations, laptops, tablets and mobile devices (ex. Android, iPhone or iPad) and even network elements.

To deploy TSM enabled-clients such as softphones, SIP-enabled iOS/Android applications or contact center agent applications, customers and 3rd party ISVs will need to incorporate the open source TSM software libraries into their applications which will establish tunnels to the TSM server.

TSM Tunnel

The following diagram briefly explains the various IP addresses utilized during the TSM session.

- TSCF External IP—This IP address is visible to any endpoint on the Internet and is used to initiate the TSM session between the TC and the TSCF. This may be configured under **security**, and then **tscf**, and then **tscf-interface**. See the TSCF Essentials Guide to configure the TSCF function on the server.
- TC Source IP—This IP address corresponds to the source address of the TC in its respective access network or it could be the IP of the Proxy behind which it is located.
- Internal Tunnel IP—This IP address will be assigned to the TC (once TLS authentication is successful) from a configured pool of IP addresses on the TSCF. It will be used to facilitate communication with the core (P-CSCF). The address pool can be configured under **security**, and then **tscf**, and then **tscf-address-pool**.
- TC Application IP—This is the IP address associated with the respective application (SIP / RTP / other) at the TC.



SDK Host Operating System Relationship

The following illustrations depict the relationship between the SDK and the host operating system:

Figure 1-2 SDK/Host OS Relationship (Simplified View)

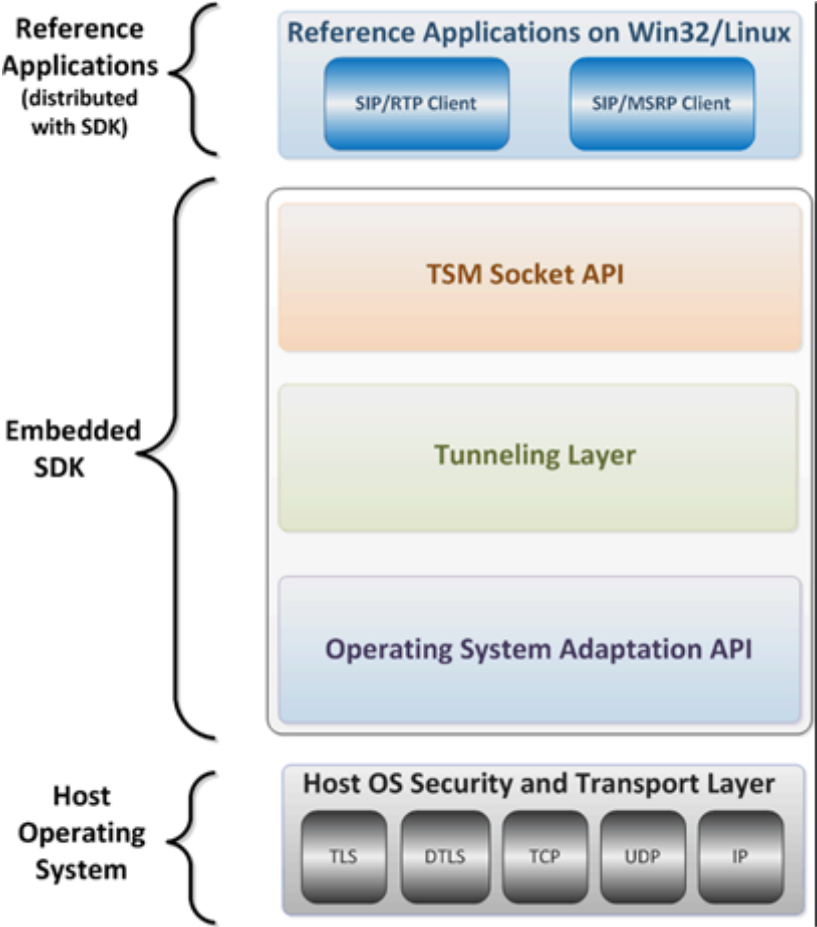
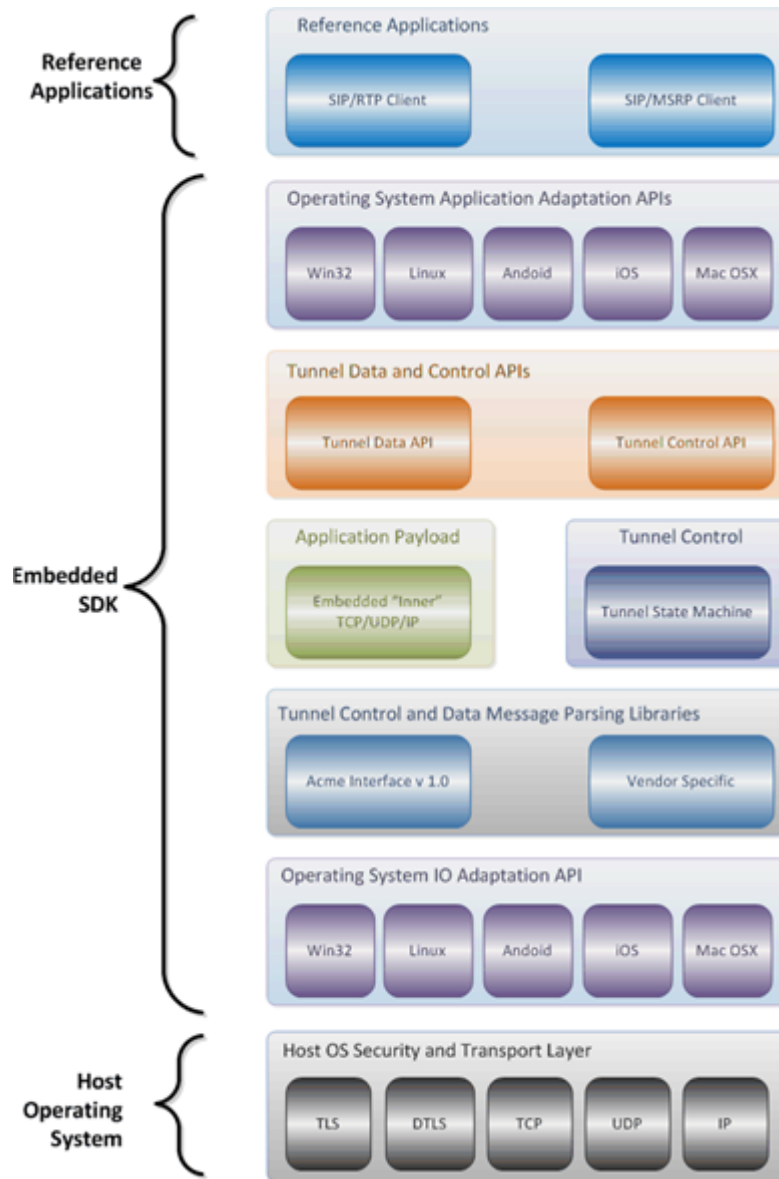


Figure 1-3 SDK/Host OS Relationship (Detailed View)



Provided Functionality

Operating Systems Support

This SDK release supports the following operating systems:

- Windows
 - Win32

 **Note:**

Windows 10 is not supported.

- Linux
- OS X
- iOS
 - 9.3
 - 10.0
 - 10.1
 - 10.2
- Android (32-bit)
 - 4.0.X (Ice Cream Sandwich)
 - 4.4 (KitKat)
 - 5.X (Lollipop)
 - 6.X (Marshmallow)

 **Note:**

64-bit Android is not supported.

Platform Support

Users of Oracle Communications Session Border Controller version S-CZ8.0.0 and higher should refer to the release notes for any platform limitations.

For releases older than S-CZ8.0.0, this SDK supports the following software and hardware combinations:

- Support for Oracle Communications Session Border Controller version S-CZ7.4.0 running on the following hardware:
 - Acme Packet 4600
 - Acme Packet 6100
 - Acme Packet 6300
- Support for Oracle Communications Tunneled Session Controller version S-CX6.4.6F6 running on the following hardware:
 - Acme Packet 4500
- Support for Oracle Communications Unified Session Manager version S-CZ7.3.5 running on the following hardware:
 - Acme Packet 4600
 - Acme Packet 6100
 - Acme Packet 6300

Proxy Support

This SDK release supports the following proxy authentication types:

- Basic
- Digest
- NTLMv2
- SPNEGO

If proxy authentication is enabled, the SDK will try to use SPNEGO authentication. If that fails, the SDK tries to use NTLMv2.

Additional Features

This SDK release also supports:

- On-the-fly integration of downloaded OpenSSL with TSCF libraries.
- Server Assigned Configuration mode
- Security Traversing Gateway (STG)
- Payload multiplexing within a tunnel
- Each SDK instance can support:
 - Up to 3 concurrent voice calls
 - Up to 10 MSRP chat sessions
 - 1 MSRP file transfer session
- Tunnel Transport
 - TCP
 - UDP
 - TLS
 - DTLS
- IP version
 - IPv4
 - IPv6

 **Note:**

When used in Decoupled Mode, the TSCF also supports mixing IPv4 and IPv6. For example, you can use an IPv6 external address outside the tunnel and an IPv4 address inside the tunnel, or vice versa.

2

Compiling the TSM Library and Documentation

Read the documentation that corresponds to your application's target operating system.

Operating System	Description	Location
	This file provides information on how to compile the TSM SDK for Android.	<code>sdk/lib/README.android</code>
	This file provides information on how to compile the TSM SDK for iOS.	<code>sdk/lib/README.iOS</code>
	This file provides information on how to compile the TSM SDK for Linux.	<code>sdk/lib/README</code>
	This file provides information on how to compile the TSM SDK for Windows. Note that Windows 10 is not supported.	<code>sdk/lib/README.WIN</code>

**WARNING:**

The OpenSSL library must be downloaded before proceeding with development.

SDK Directories

SDK directories are shown below. Note that not all listed directories may be present (or supported) in the current release.

Path	Description
apps	SDK based applications
apps/linphone	Open source VoIP application utilizing TCP/TLS/DTLS/UDP for tunnel transport that has been ported to the SDK. Platform support limited to Windows, Android, and iOS. Refer to www.linphone.org for additional details.
apps/sipp	Open source SIP traffic generator application utilizing TCP tunnel transport that has been ported to the SDK. Platform support limited to Linux. Refer to www.sourceforge.net for additional details.
apps/tsc_sip	Reference demonstration/development guide app (tsc_sip_client.c)
docs	SDK Documentation
docs/html	Authoritative API HTML-based documentation. Access via ".../html/index.html" after running `make doxygen`.
extlib	External, optional libraries
lib	SDK Library source – to be linked with the target application
lib/android-ndk	Android Specific library instructions and precompiled libs
lib/CSM	tunneling Client State Machine
lib/EIP	Embedded TCP/UDP/IP Stack
lib/include	SDK API definitions
lib/OSAA	Operating System Application Adaptation APIs
lib/TAPI	Tunnel Data and Control APIs
lib/TPL	Tunnel Control and Data Message Parsing Libraries
tools	Development Tools
tools/wireshark	TSCF protocol dissector

Download and Compile OpenSSL

The default version of OpenSSL has been removed. Developers should download the desired version of OpenSSL and modify the build script to allow on-the-fly integration with the SDK.

1. Download the version of OpenSSL you want to integrate into the SDK.
The customer is responsible for selecting a secure version of OpenSSL from <https://www.openssl.org/>. Oracle can confirm OpenSSL version 1.0.1g works with the SDK.
2. In the build script for your target operating system, set the `VERSION` variable to the version number of OpenSSL.

The build scripts are located in the `sdk/extlib` directory and the `VERSION` variable is found at the top of the script.

```
VERSION="1.0.1g"
```

3. If compiling version 1.0.1g for iOS, run the `patch-openssl-1.0.1g` script.

```
./patch-openssl-1.0.1g
```

4. Run the build script for your target operating system.

For example:

```
./build_androidlib.sh
```

Generate the API Documentation

1. Navigate to the `lib` folder.

```
cd sdk/lib
```

2. Generate the API documentation.

```
doxygen ../docs/doxygen/doxygen.conf
```

3. Open the documentation using a browser pointing to `sdk/docs/html/index.html`.

3

Accessing and Using the TSM SDK APIs

Sample TSM SDK-based Applications

A number of small application templates are found throughout the apps directory. Each of these files contains a small, well-defined set of functionality that enables a software developer to easily understand its implementation via the TSCF client-side SDK.

SDK:	
+---apps/tsc_sip/tsc_sip_client +---tsc_sip_client.c	Provides a basic SIP client utilizing TCP/TLS/UDP/DTLS for tunnel transport of inner UDP sockets. Supports Linux, Windows, and Android operating systems.
+---apps/tsc_sip/tsc_sip_server +---tsc_sip_server.c	Provides a basic SIP server utilizing TCP/TLS/UDP/DTLS for tunnel transport of inner UDP sockets. Supports Linux, Windows, and Android operating systems.
+---apps/tsc_sip/tsc_sip_inner_tcp +---tsc_sip_inner_tcp.c	Provides a basic SIP client utilizing TCP for tunnel transport and demonstrating usage of TCP sockets for applications such as HTTP -- supports Linux operating systems.

All of the above files contain extensive comments making it an easy task to navigate through the code. Using `tsc_sip_client.c` as an example, you can readily proceed through the file.

1. Search for `tsc_ctrl_init ()` and examine the code immediately following this function for the details of tunnel initialization.
2. Search for Create a Tunnel and examine the code immediately following for the details of tunnel creation and the configuration exchange between the TSCF server and client.
3. Search for SIP SOCKET CREATION and examine the code that creates and binds TSCF sockets.
4. Search for REGISTER TRANSACTION and examine the code that builds a SIP REGISTRAR request and processes the REGISTRAR response.
5. Search for INVITE TRANSACTION and examine the code that builds a SIP INVITE request and processes the REGISTRAR response.
6. Search for Build and send ACK to examine ACK creation and processing code.
7. Search for RTP Exchange to examine RTP code.
8. Search for BYE TRANSACTION to find code that terminates a SIP connection.
9. Search for TEST DONE to find code that terminates a tunnel.

Using The SDK To Create A TSM Tunnel

The following steps provide an outline on integrating a SIP client (in this example `tsc_sip_client`) with the TSM SDK. Please refer to the file `tsc_sip_client.c` (located at `sdk/apps/tsc_sip/tsc_sip_client`) which contains working code references on establishing a TSM tunnel and making a SIP/RTP based call.

Initialize the TSCF-Client side API

Use the TSCF Server IP address, port, transport type, wireshark tracing and certificate parameters (if using TLS/DTLS) to initialize the client. Populate the required information in a `tsc_tunnel_params` type structure.

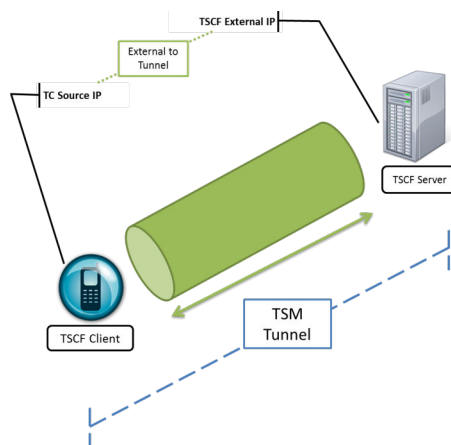
Search for "`tsc_ctrl_init()`" in the reference file for actual code implementation.



Create a TSM Tunnel

Once initialized, create a TSM tunnel between the TSCF client and server, register for callbacks and obtain the SIP server IP address.

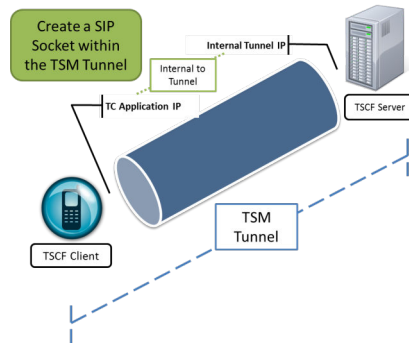
Search for the text "Create a tunnel" in the reference file for actual code implementation.



Create a SIP Socket within the TSM Tunnel

With the TSM tunnel established, create a SIP socket and bind the local address assigned by the TSCF server to it.

Search for the text "SIP SOCKET CREATION" in the reference file for actual code implementation.



Send SIP messages over the TSM Tunnel

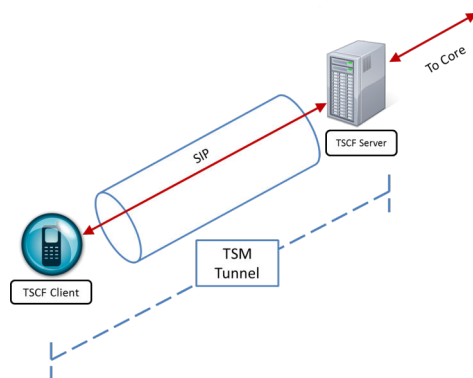
You can now send SIP messages over the TSM tunnel. Based on the kind of SIP applications, you may need to send a REGISTER message or directly initiate a peer-to-peer call through an INVITE message.

You can find examples of both messages being sent by looking for the following text. Since this is only sample code, please use the same as reference for sending and receiving SIP messages via the TSM tunnel.

REGISTER TRANSACTION: Constructs and sends a REGISTER message to the tsc_sip_server. The code currently doesn't check for a 200 OK message.

INVITE TRANSACTION: Constructs and sends an INVITE message and handles a 200 OK.

Build and send ACK (to INVITE transaction) : This code builds and sends an ACK to the INVITE received.

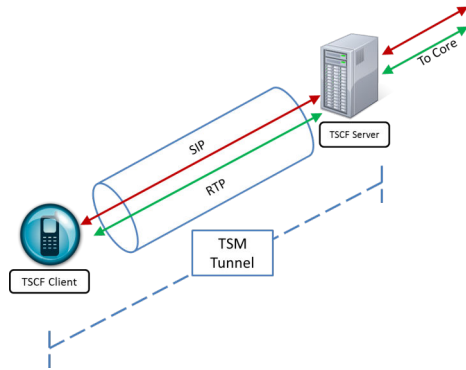


Send RTP message over the TSM Tunnel

As part of the call, there is also sample code to send and receive RTP packets over the TSM tunnel. Search for the text "RTP EXCHANGE" within the reference file for a code implementation.

The code builds RTP packets purely for the purpose of simulation.

At this point you should be able to make calls through the TSM tunnel.



Terminate a SIP call over the TSM Tunnel

After you are done with the call, you may terminate it using sample code in the reference file. Search for the text "BYE TRANSACTION".

Terminate a TSM Tunnel

If your application has determined that it doesn't require the use of a TSM tunnel, you may terminate the tunnel using the sample code. Search for "Finished, Socket, Tunnel Cleanup".

Enabling Redundancy

TSM enables an application to improve media quality under adverse network packet loss through the tunnel redundancy feature.

Once enabled, tunnel redundancy comes with two options:

1. Original TSM tunnel + 1 redundant tunnel
2. Original TSM tunnel + 2 redundant tunnels

Increasing the number of tunnels increases media quality at the cost of network bandwidth. Decide what kind of redundancy you require before enabling this feature.

Tunnel redundancy can be enabled on a per-socket basis.

1. Create a notification handler function. Once requested, the TSCF notifies the application whether the redundancy was enabled successfully.
2. Set a socket option with the type of redundancy factor (1 or 2).

Search for the text "RTP socket created" and "TSC_REDUNDANCY" in the reference file for actual code implementation that creates redundant tunnels for RTP packets.

Error Codes

The following are error codes you can check against when calling the TSCF layer to ensure you can handle all success and failure scenarios:

Return Value	TSC_ERROR_CODE
0	tsc_error_code_ok
1	tsc_error_code_error

Return Value	TSC_ERROR_CODE
2	tsc_error_code_not_logged
3	tsc_error_code_cannot_connect
4	tsc_error_code_cannot_configure
5	tsc_error_code_keepalive_failure
6	tsc_error_code_service_failure
7	tsc_error_code_cannot_recv_data
8	tsc_error_code_no_data
9	tsc_error_code_cannot_send_data
10	tsc_error_code_authenticate
11	tsc_error_code_cannot_release
12	tsc_error_code_queue_overflow
