

# SSL/TLS on Symbian

## TracNav

- **Getting Started**
  - ◆ **Preparation**
    - ◇ [Get the source code](#)
    - ◇ [Disk Space Requirements](#)
    - ◇ [Build Preparation](#)
  - ◆ **Build for Desktop**
    - ◇ [Windows](#)
    - ◇ [Linux](#)
    - ◇ [MacOS X](#)
    - ◇ [Python](#)
  - ◆ **Build for Mobile**
    - ◇ [iOS: Apple iPhone, iPad, and iPod Touch](#)
    - ◇ [Android](#)
    - ◇ [BlackBerry 10 \(BB10\)](#)
    - ◇ [Windows Mobile](#)
    - ◇ [Windows Phone 8.x and UWP](#)
    - ◇ **Symbian**
      - [Using Audio Proxy Server \(APS\)](#)
      - [Using VoIP Audio Services \(VAS\)](#)
      - [Using Transport Layer Security \(TLS\)](#)
  - ◆ [Build for Other](#)
  - ◆ **Next: Using the libraries**
    - ◇ [Running pjsip Applications](#)
    - ◇ [Building Application using PJSIP with GNU Tools](#)
    - ◇ [Video User's Guide \(2.0 onwards\)](#)
- **See Also**
  - ◆ [Installing OpenSSL on Windows](#)
  - ◆ [Using Subversion](#)
  - ◆ [Visual Studio Build Configurations](#)
  - ◆ [Using Eclipse with PJSIP](#)
  - ◆ [S60 3rd Edition devices](#)

## Table of Contents

PJSIP provides secure communications via secure socket abstraction, `pj_ssl_sock_*`, which can be used by the higher level applications, such as SSL/TLS SIP transport to perform secure SIP signaling. On Symbian platforms, the secure socket implementation is done natively using `CSecureSocket` class. This feature is available from version 1.5 onwards.

## Scope

Secure socket implementation on Symbian provides:

1. Transparent SSL/TLS operations, application uses the secure socket basically the same way as using normal socket, e.g: when connection completion status is reported (via callback) as successful, it means that both the underlying socket connection and the SSL/TLS handshake are successful.
2. Active socket operations as provided by [http://www.pjsip.org/pjlib/docs/html/group\\_PJ\\_ACTIVE\\_SOCKET.htm](http://www.pjsip.org/pjlib/docs/html/group_PJ_ACTIVE_SOCKET.htm) Active Socket I/O.
3. List of trusted Certificate Authorities (CA) is based on Symbian Certificate Management, e.g: in E65, Main Menu > Tools > Settings > Security > Certificates Management.
4. Support for SSL 3.0 and TLS 1.0.

## Limitations

1. Only support for client mode (CSecureSocket limitation).
2. Specifying client credential (e.g: certificate and the corresponding private key) is not supported (CSecureSocket limitation), so secure socket may not be able to connect to server that requires client certificate.
3. Currently, server certificate verification is only done internally by CSecureSocket, further verification mechanism by application (e.g: via callback) is not supported. **Note** that untrusted server certificates result in a user dialog.
4. Managing (adding/editing/deleting) entry of trusted CA list should be handled by application.

## Enable SIP transport SSL/TLS on symbian\_ua sample application

1. Enable TLS as described [here](#).
2. Modify transport setting in ua.cpp:

```
#define ENABLE_SIP_TLS 1 // default is 0
```

3. You have to set the SSL/TLS server name field accordingly, otherwise the connection will either fail with !KErrAbort/Interrupted or a warning dialog about different server name will be displayed:

```
#define TLS_SRV_NAME "pjsip.org"
```

4. Update other related configurations ua.cpp such as SIP account, e.g:

```
#define HAS_SIP_ACCOUNT 1
#define SIP_DOMAIN "your_domain/realm"
#define SIP_USER "your_userid"
#define SIP_PASSWD "your_pass"
#define SIP_PROXY "<sip:some_proxy;transport=tls;lr>"
```

**Note** that without registering an account into a registrar, symbian\_ua will not be able to be contacted (e.g: receive calls), as the secure socket backend (CSecureSocket) can only work as client.

1. If you don't use SIP account (for example for quick testing only), don't forget to add ";transport=tls" parameter to your destination URI, e.g.:

```
#define SIP_DST_URI "<sip:100@pjsip.org;transport=tls>"
```

# Building your own application using SSL/TLS on Symbian

1. Enable TLS as described [here](#).
2. If the **low level** secure socket is needed, include `ssl_sock.h`:

```
#include<pj/ssl_sock.h>
```

3. When using **PJSUA-LIB**, SIP transport TLS can be enabled by instantiating SIP transport type `PJSIP_TRANSPORT_TLS`, e.g (captured from `symbian_ua ua.cpp`):

```
pjsua_transport_config tcfg;
pjsua_transport_id tid;

pjsua_transport_config_default (&tcfg);
tcfg.port = SIP_PORT;
tcfg.tls_setting.server_name = pj_str(TLS_SRV_NAME);
status = pjsua_transport_create(PJSIP_TRANSPORT_TLS, &tcfg, &tid);

// then, specify "transport=tls" URI param in the proxy/registrar URI,
// e.g: "<sip:some_proxy;transport=tls>"
```

4. **Link** the application to `securesocket.lib`, by specifying the library in the application MMP:

```
LIBRARY securesocket.lib
```

## Troubleshooting

### Error -7547

```
0.853 pjsua_acc.c Registration sent
2.279 tlsc0x2432b71c TLS connect() error: Symbian native error -7547 [code=127547]
2.294 tsx0x243184b8 Failed to send Request msg REGISTER/cseq=38313 (tdta0x24328cf0)! err=127547 (Symbian nati
2.304 pjsua_acc.c SIP registration failed, status=503 (Symbian native error -7547)
2.314 symbian_ua.cpp Registration failed!
2.431 tlsc0x2432b71c TLS transport destroyed with reason 127547: Symbian native error -7547
```

Symbian error -7547 is about set `setOpt`. Check the source code, you have to define `servername`.

### Error `KErrAborted` / Interrupted

You need to set the `servername` field.