Network connections guide for JC-E SECURE converters

(document version 1.31)

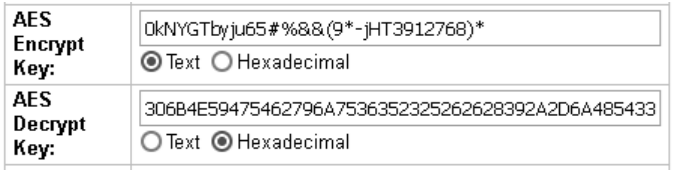
Table of Contents

1. Methods of securing M-Bus communication - secure tunnel connection (M-Bus port and Ethernet)

1.1. AES

AES (Advanced Encryption Standard) symmetric block cipher algorithm
The key length can be 128, 192, or 256 bits. It is designed to encrypt blocks of data. AES ensures confidentiality (encryption), not data integrity or authenticity. It is possible to use one key for encryption (outgoing data) and another key for decryption (incoming data), which increases encryption security. The TCP protocol is supported in server mode. In client mode, TCP and UDP protocols are supported. The key can be entered as ASCII text or in hex format.

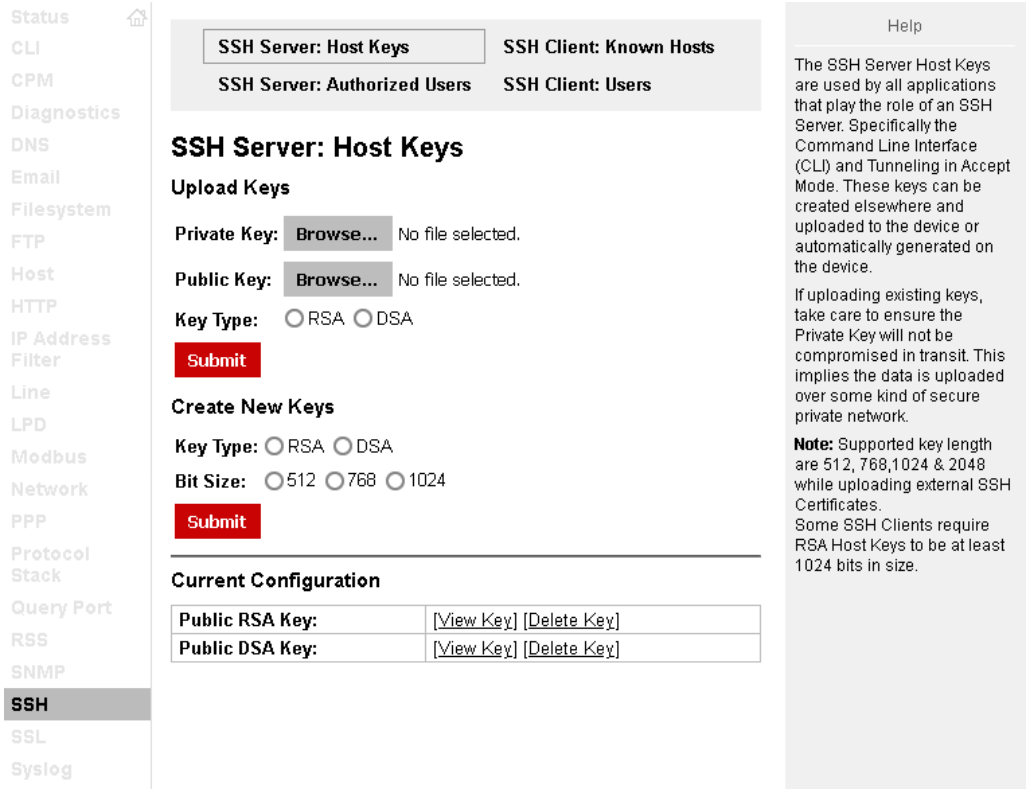
The settings for this type of connection can be found in the converter's webinterface
if the converter is going to act as a server under Tunnel > Accept Mode > Protocol, or
if the converter is going to act as a client under Tunnel > Host 1 > Connect Mode > Protocol.

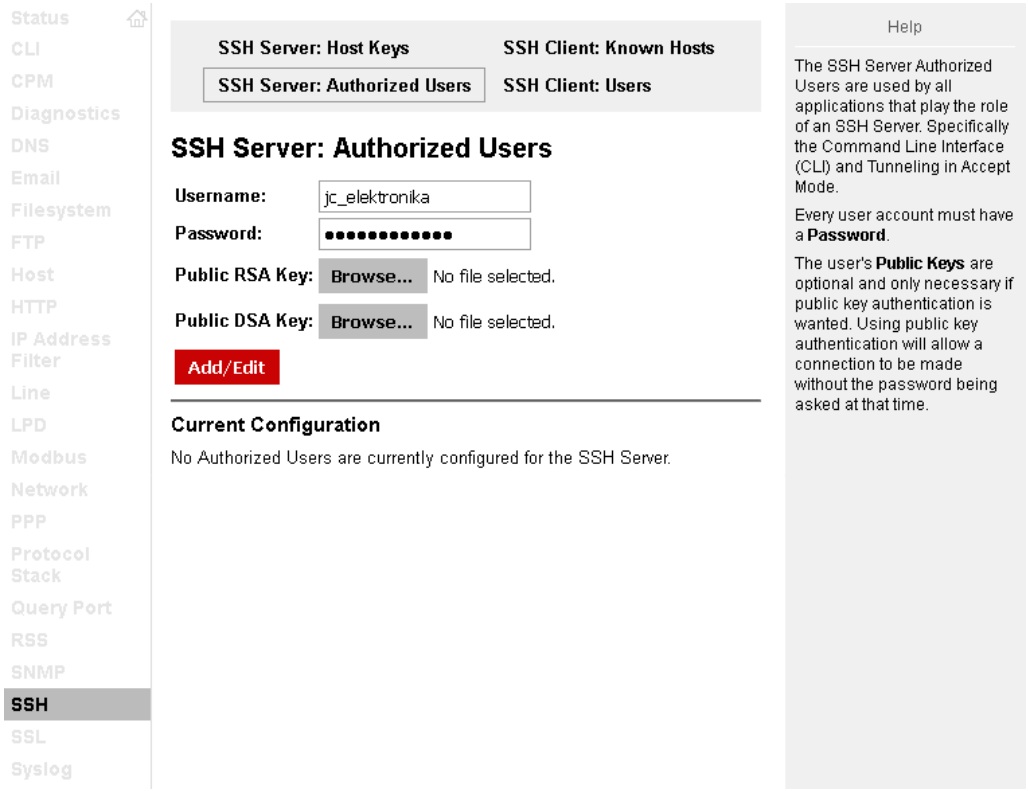


1.2. SSH

SSH is one of the most reliable and common methods of securing TCP connections. It is primarily designed to secure the HTTPS protocol, but it is also possible to tunnel M-Bus data via a TCP connection. It ensures confidentiality (encryption), integrity, and authenticity of data.

Server mode - user key(s) must be uploaded and at least one SSH authorized user must be defined. Keys can be generated in the converter or uploaded via the webinterface.
RSA and DSA keys with a length of 512, 768, 1024 bits are supported. The user must have a defined username and password. User public keys are optional and only necessary if public key authentication is required. Using public key authentication allows you to establish a connection without entering a password in this mode.

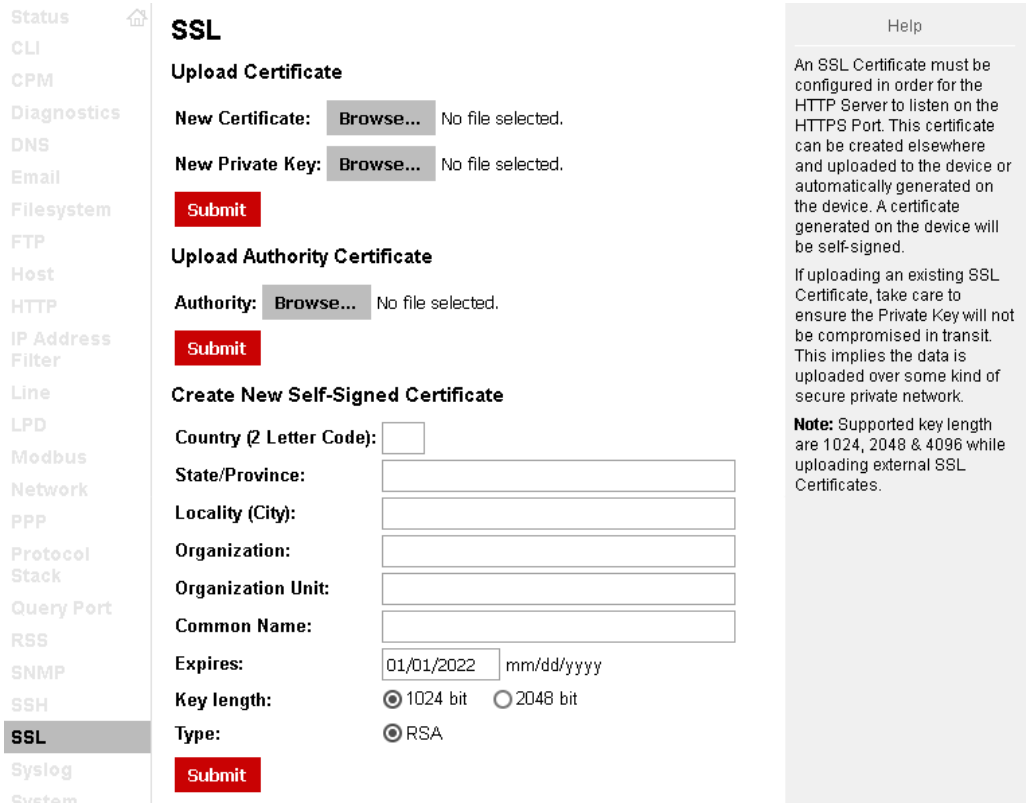




Client mode - setting the public keys is optional, but if they exist, they provide an additional layer of security that helps prevent Man-in-the-Middle (MITM) attacks.
When adding host public keys for the server, enter either the DNS name of the server or its IP address. The server name should match the tunneling settings.
At least a password or a pair of keys must be configured for the user. Keys for authentication can be generated in the converter or uploaded via the web interface.

1.3. SSL

SSL is a universal encryption protocol for securing TCP/IP communication. The converter can use it to encrypt HTTPS (web pages) or tunnel M-Bus data.
It ensures the confidentiality (encryption), integrity, and authenticity of the transmitted data.
The converter supports TCP/IP tunneling in both server and client modes.
It is possible to generate and use a self-signed certificate in the converter or you can upload your own certificate together with an authorization certificate. The self-signed certificate is of the RSA type with a key length of 1024 or 2048 bits.



2. Secure connections

There are multiple ways of creating a secure connection using the converter, this guide covers only the following three methods:

1. TCP AES connection with the converter as a server and a virtual serial COM port application as a client.
2. SSL connection with the converter acting as a client.
3. SSL connection with the converter acting as a server.

To create an SSL connection an application called stunnel for Windows will be used.

The stunnel installer can be downloaded here:
https://www.stunnel.org/downloads.html

Stunnel basic information:
https://www.stunnel.org/howto.html

Detailed information on stunnel configuration options:
https://www.stunnel.org/static/stunnel.html

It is not required but an open source tool called TCP/IP builder can be used to test and troubleshoot the connection chain. More details here:
https://www.drk.com.ar/en/legacy/tcp-ip-builder/

2.1. TCP AES with converter server and virtual COM port client

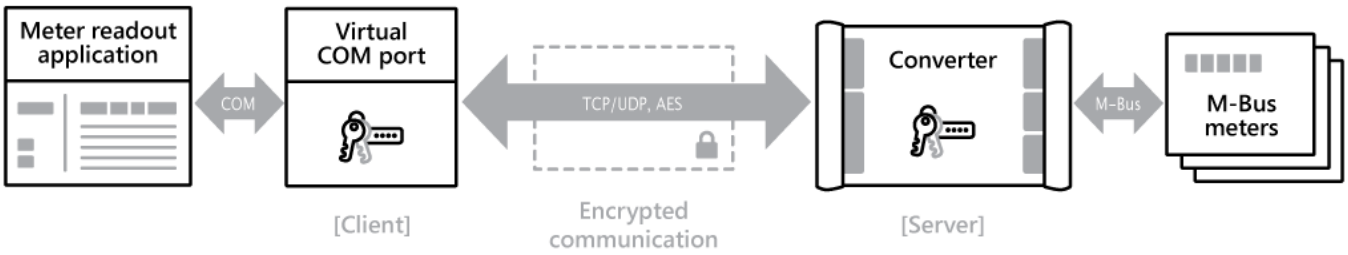


Diagram of a TCP AES connection with converter server and a secure virtual COM port client.

Same encryption key is used on both sides.

⚠ **Important note:**
A firewall used on Windows or anywhere in the connection chain must be properly configured so that the connection is not blocked. If a connection cannot be established despite correct settings firewall configuration should be thoroughly examined.

2.1.1 Converter server with TCP AES

Configuration steps:

1. Open the converter webinterface in a web browser.
   The default IP is 169.254.100.10
2. Open the **Tunnel** settings in the left sidepanel.
3. Open the **Accept Mode** setting
4. Set the following settings:

   Mode: Always

   Local port: 10001 (or any desired port number)

   Protocol: TCP AES

   AES Encrypt Key: Enter a key in hex format, 128 bit long, 32 hex chars. without spaces, e.g. 00112233445566778899aabbccddeeff

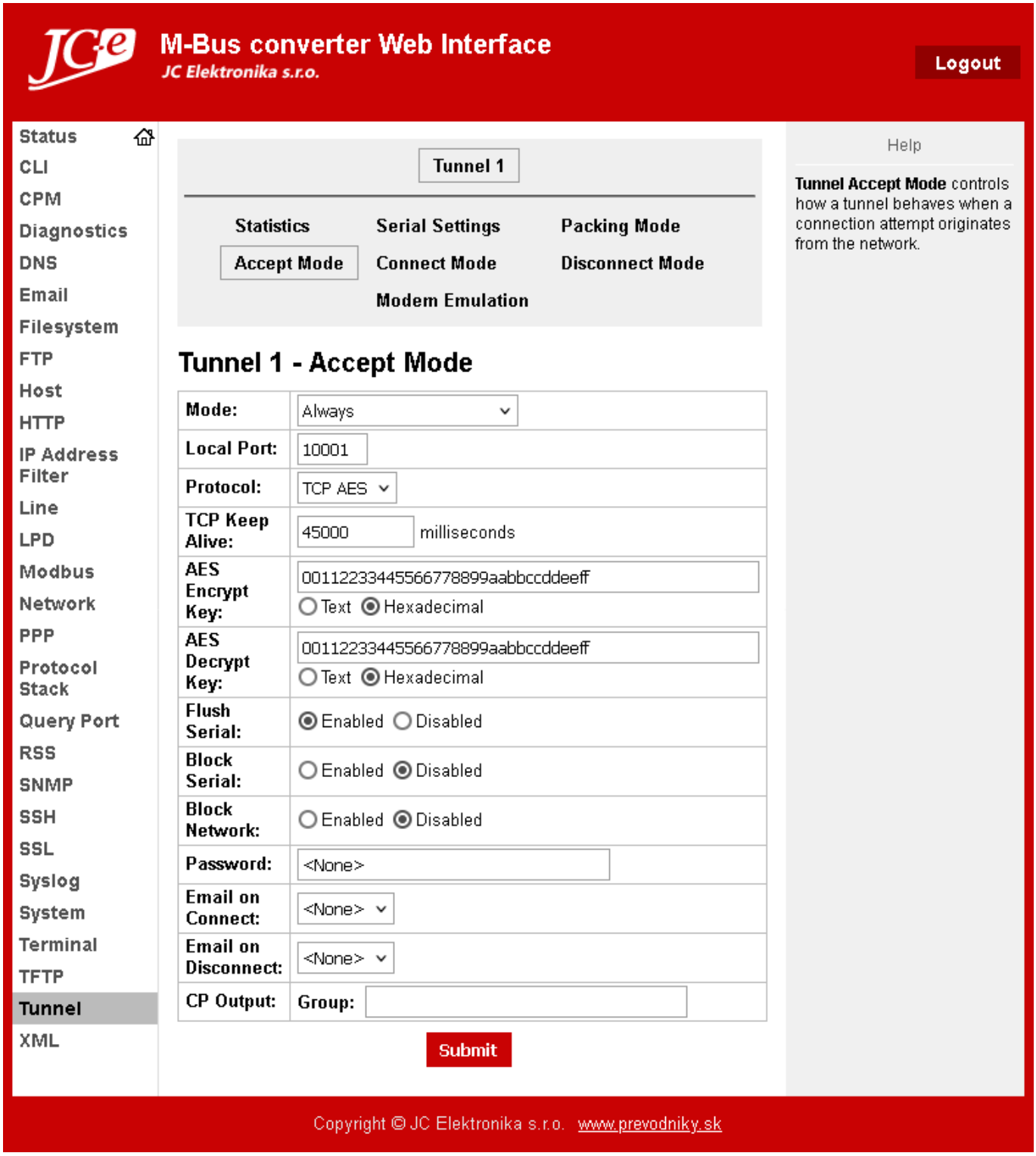
   Format: ◉ Hexadecimal

   AES Decrypt Key: Enter the same key as the Encrypt Key

   Format: ◉ Hexadecimal

   The remaining settings do not need to be changed unless desired.

5. Click **Submit**.



2.1.2 Lantronix Secure Com Port Redirector client

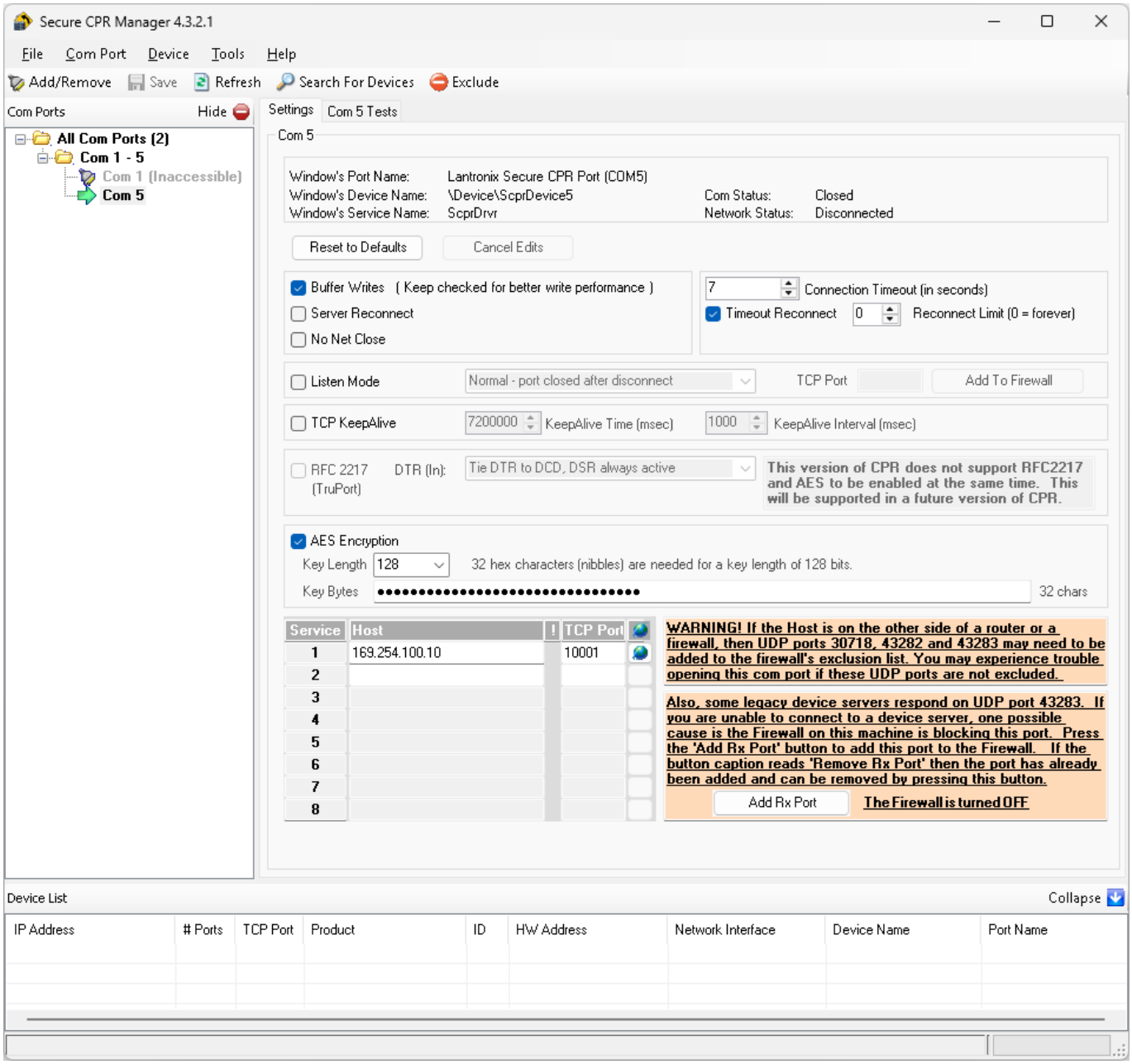
To create a virtual serial port we will be using the Lantronix Secure Com Port Redirector application.
It can be downloaded here:
https://papouch.com/lantronix-secure-com-port-redirector-p7194/

Configuration steps:

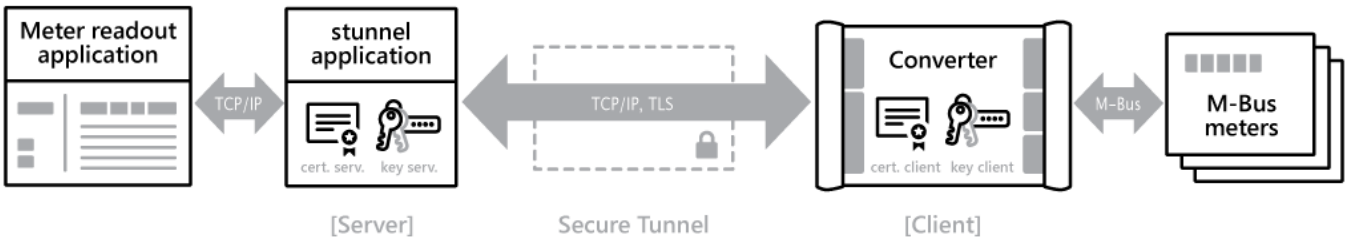
1. Open Lantronix Secure Com Port Redirector.
2. Click the **Add/Remove** button on the toolbar to add a COM port.

3. Select the created port e.g. **COM 5** in the left sidepanel.
4. Check the **AES Encrytion** checkbox.
5. Set **Key length:** 128bits.
6. Set **Key bytes:** Enter a key in hex format, 32 hex chars. without spaces,
   e.g. 00112233445566778899aabbccddeeff

   The key must match the encrypt and decrypt key used by the converter.
7. In the Service table set one row
   Host: 169.254.100.10 (IP address of the converter)
   TCP Port: 10001 (port of the converter)
8. In the top toolbar click the **Save** button.



2.2. SSL with converter client and stunnel server



SSL connection with converter as client and stunnel as server.

The converter should use its own certificate and private key. The stunnel application or any application taking its place like a SCADA system should also use its own certificate and private key that are different from the converter side.

Note that using a secure connection adds a 5ms delay to the data communication.

⚠ **Important note:**
A firewall used on Windows or anywhere in the connection chain must be properly configured so that the connection is not blocked. If a connection cannot be established despite correct settings firewall configuration should be thoroughly examined.

2.2.1 Converter client with SSL

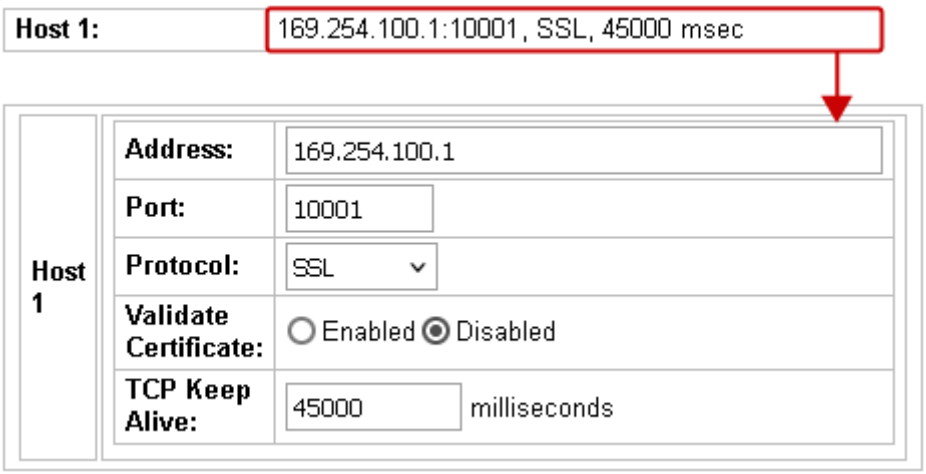
Steps to configuring the converter as a client:

1. Open the converter webinterface in a web browser.
   The default IP is 169.254.100.10
2. Set an SSL certificate and a private key in the SSL configuration.
   See chapter: Configuring SSL.
3. Open the **Tunnel** settings in the left sidepanel.
4. Open the **Connect Mode** settings.
5. Set the following:

   Mode: Always

   Local Port: Leave empty.

   Host 1: Click on the table cell to open the detailed settings for Host 1



For Host 1 set:

Address: 169.254.100.1 (IP address of the stunnel server)

Port: 10001 (port of the stunnel server)

Protocol: SSL

Validate Certificate: Disabled

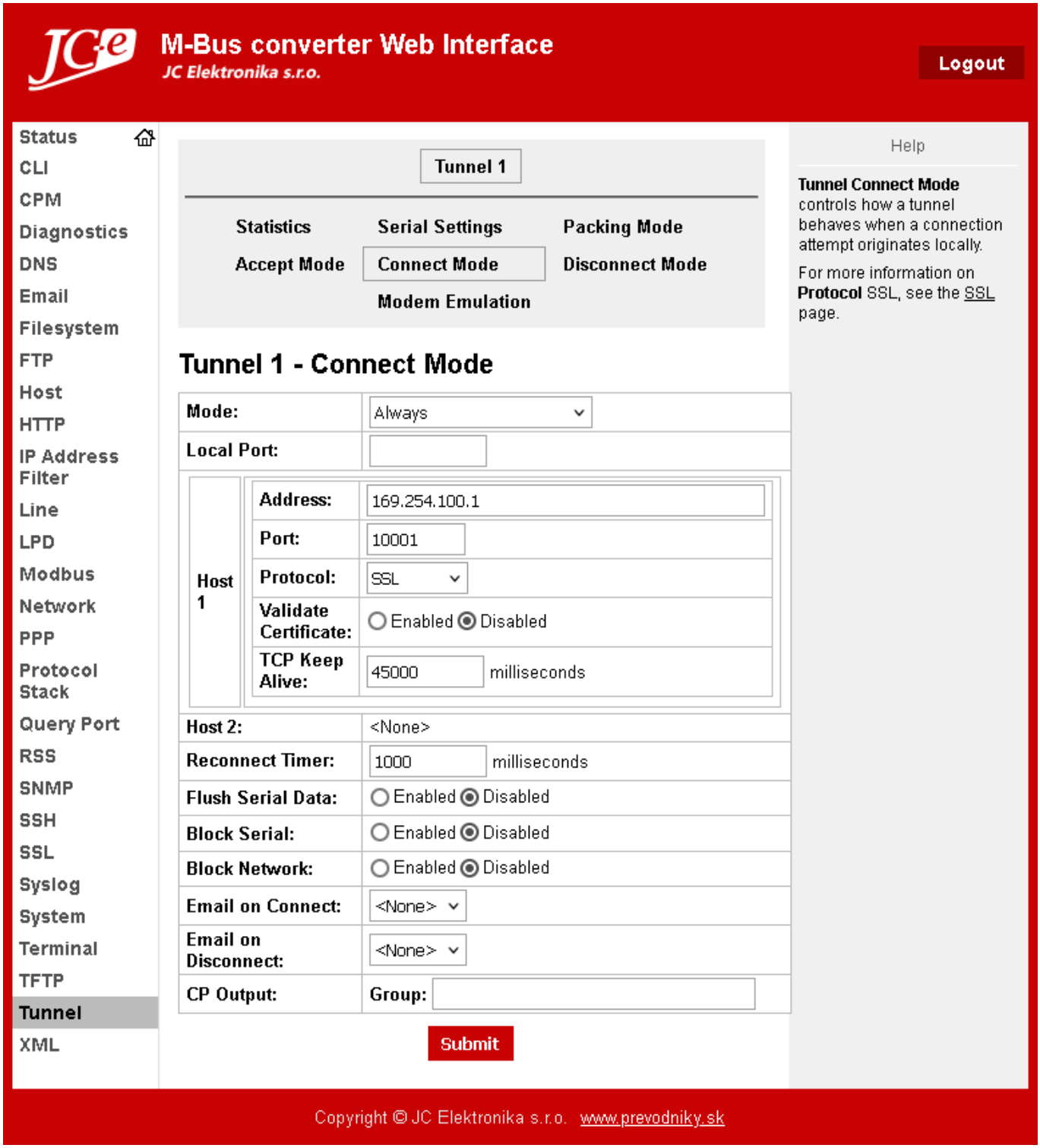
Note:
If you want to enable certificate validation you will need to go to the **SSL** settings and in the **Upload Authority Certificate** section add the certificate used by stunnel to the Certificate Authority.

The stunnel server certificate (`cert_stunnel.pem`) needs to be uploaded there so that the converter can use it for validation. Multiple certificates can be uploaded to the converter into its certificate store. This is done by selecting a single certificate file, clicking Submit and repeating this process for multiple files.

The remaining settings do not need to be changed unless desired.

6. Click **Submit**.

   The converter will periodically try to create a connection to the server at the Reconnect Timer interval, 1 second by default.



2.2.2 Stunnel server with SSL

We will be using the stunnel application to create the server. See introduction in Secure connections.

Steps for configuring stunnel server:

1. Create a private encryption key and a certificate for use in stunnel.
   See chapter: Creating a private key and a self-signed certificate.
   If you already created these during the converter configuration you can continue to step 2.

2. Open stunnel using the shortcut in the Windows start menu **stunnel GUI start**.

3. Edit the stunnel configuration. From main menu choose *Configuration* > *Edit configuration*.

   The application comes with a sample configuration in the stunnel.conf file. Comment out all sections using semicolon (;) at the start of each line or backup the file and create a completely new stunnel.conf file containing just your own configuration.

4. Add the following lines to the configuration file, change the IP and port values according to your needs.

```
[stunnelserver]
accept = 169.254.100.1:10001
connect = 127.0.0.1:10002
cert = cert_stunnel.pem
key = key_stunnel.pem
```

5. If you need certificate validation using a Certificate Authority following lines can be added.

```
verify = 0
verifyChain = yes
verifyPeer = yes
CAfile = ca_cert_converter.pem
```

   The `ca_cert_converter.pem` is the same certificate file as used by the converter.

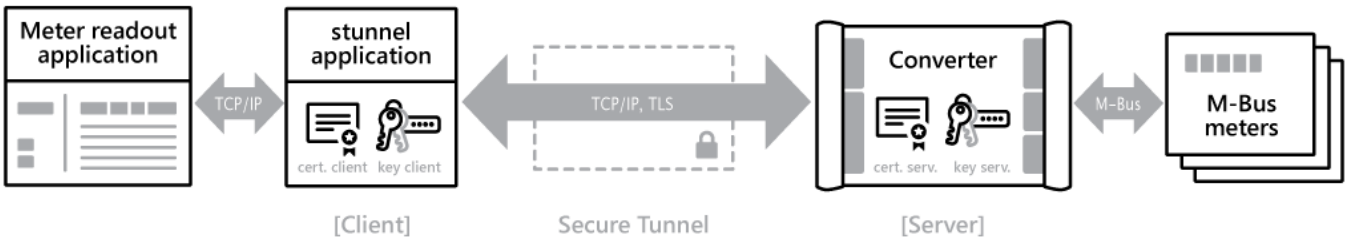
   Multiple certificates can be stored in one file or a directory containing multiple certificates can be used for validation, description for such setting is not part of this guide.

   More information about stunnel configuration options can be found here:
   https://www.stunnel.org/static/stunnel.html

6. From the stunnel main menu choose **Configuration** > **Reload configuration**.

   Stunnel should now be ready to accept the connection from the converter.

2.3. SSL with converter server and stunnel client



SSL connection with converter as server and stunnel as client.

The converter should use its own certificate and private key. The stunnel application or any application taking its place like a SCADA system should also use its own certificate and private key that are different from the converter side.

Note that using a secure connection adds a 5ms delay to the data communication.

⚠ **Important note:**
A firewall used on Windows or anywhere in the connection chain must be properly configured so that the connection is not blocked. If a connection cannot be established despite correct settings firewall configuration should be thoroughly examined.

2.3.1 Converter server with SSL

When the converter will act as a server with SSL it will require a private encryption key and certificate to be present on the converter. You can use a self-signed certificate that is generated internally by the converter or you can provide your own certificate.

When uploading an existing SSL certificate, take care to ensure the private key will not be compromised in transit. This implies the data is uploaded over some kind of secure private network.

Steps for configuring the converter server:

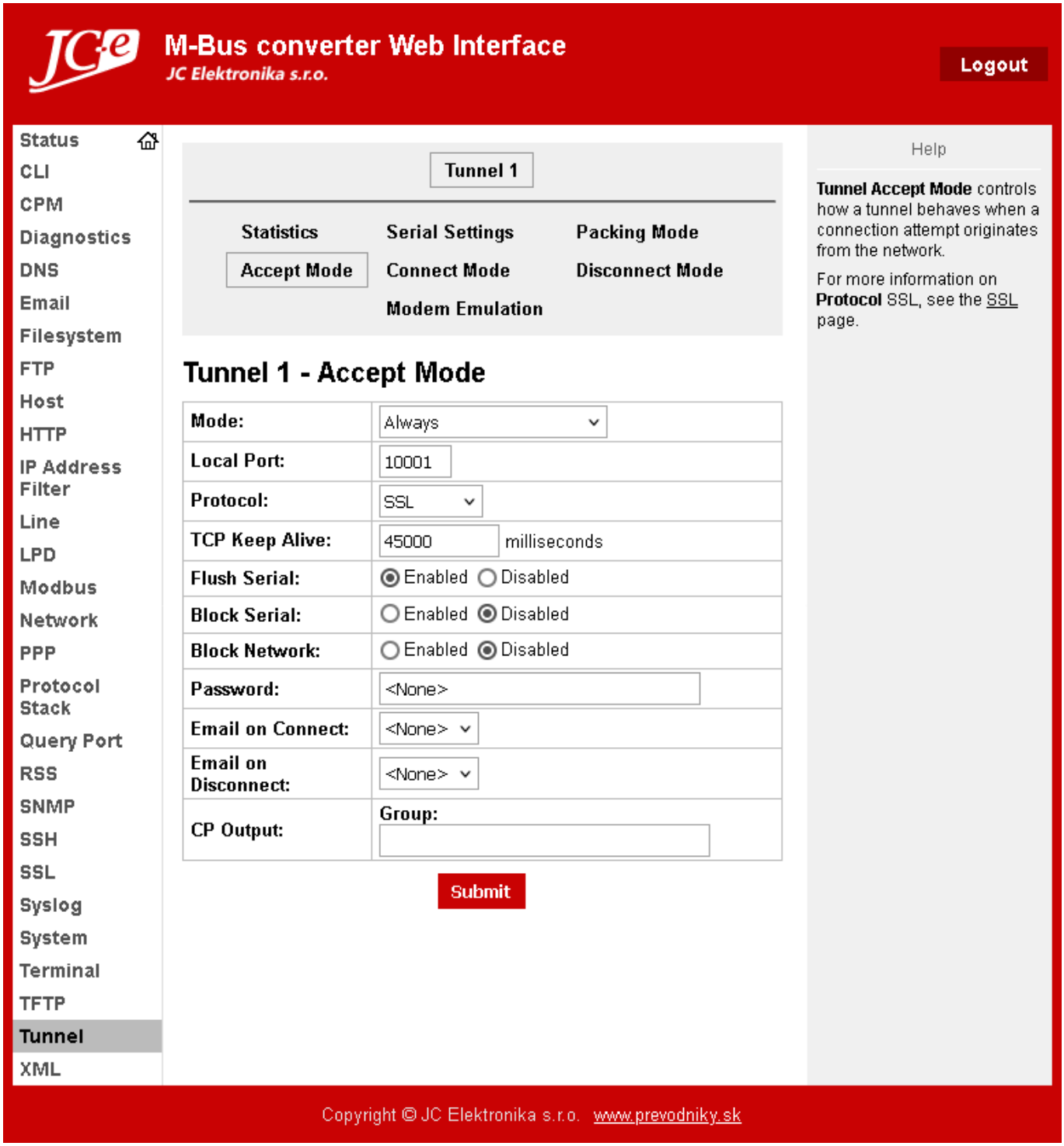
1. Open the converter webinterface in a web browser.
   The default IP is 169.254.100.10
2. Set an SSL certificate and a private key in the SSL configuration.
   See chapter: Configuring SSL.
3. Open the **Tunnel** settings in the left sidepanel.
4. Open the **Accept Mode** settings.
5. Set the following:

   Mode: Always

   Local Port: 10001 (Port where the stunnel client will connect to)

   Protocol: SSL

   The remaining settings do not need to be changed unless desired.
6. Click **Submit**.



2.3.2 Stunnel client with SSL

We will be using the stunnel application to create the server. See introduction in Secure connections.

Steps for configuring the stunnel client:

1. Create a private encryption key and a certificate for use in stunnel.
   See chapter: Creating a private key and a self-signed certificate.
   If you already created these during the converter configuration you can continue to step 2.

2. Open stunnel using the shortcut in the start menu **stunnel GUI start**.

3. Open stunnel configuration file for editing from main menu choose Configuration > Edit configuration.

   The application comes with a sample configuration in the `stunnel.conf` file.
   Comment out all sections and options using semicolon (;) at the start of each line to make a clean configuration.
   Alternatively you can backup the file and create a completely new `stunnel.conf` file containing just your own configuration.

4. Add the following lines to the configuration file, change the IP and port values according to your needs

```
[stunnelclient]
accept = 127.0.0.1:10001
connect = 169.254.100.10:10001
client = yes
verify = 0
cert = cert_stunnel.pem
key = key_stunnel.pem
sslVersion = TLSv1.2
options = LEGACY_SERVER_CONNECT
options = NO_TICKET
options = DONT_INSERT_EMPTY_FRAGMENTS
```

5. If you need certificate validation using a Certificate Authority following lines can be added:

```
verifyChain = yes
verifyPeer = yes
CAfile = ca_cert_converter.pem
```

   The `ca_cert_converter.pem` is the same certificate file as used by the converter.

   Multiple certificates can be stored in one file or a directory containing multiple certificates can be used for validation, description for such setting is not part of this guide.

   More information about stunnel configuration options can be found here:
   https://www.stunnel.org/static/stunnel.html

6. From the main menu choose *Configuration* > *Reload configuration*.

   Stunnel should now be ready to create a connection to the converter acting as a server.
   The connection with the converter will be established only after the connection between stunnel and the meter readout application is created.

3. HTTPS for converter webinterface

To enable access over HTTPS an SSL certificate and private encryption key must be configured. This certificate can be uploaded to the converter or can be generated directly on the converter.

See chapter: Configuring SSL.

After setting up SSL the HTTPS access will become available.

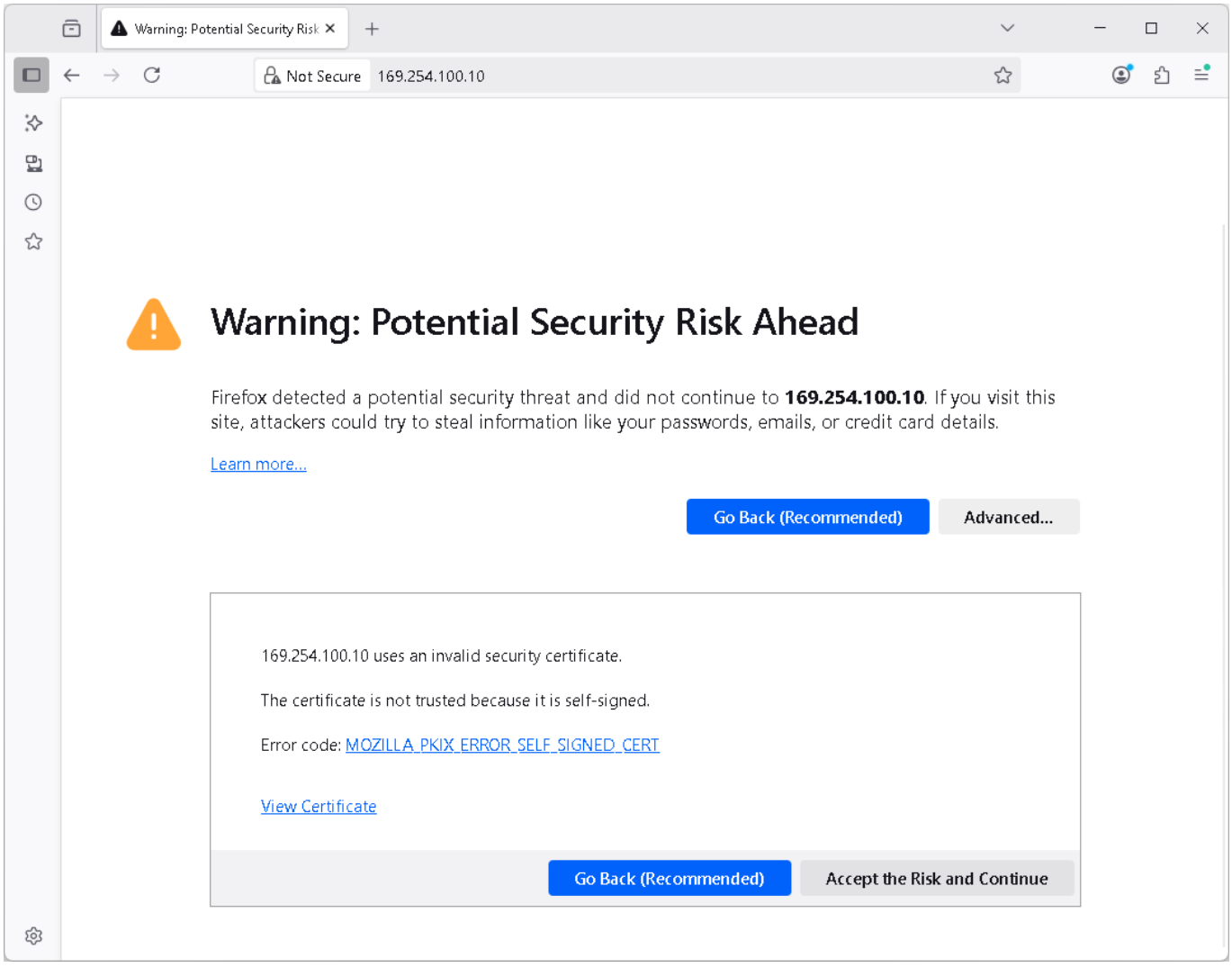
When uploading an existing SSL certificate, take care to ensure the Private Key will not be compromised in transit. This implies the data is uploaded over some kind of secure private network.

Supported key lengths are 1024, 2048 and 4096 bits with external SSL Certificates.

⚠ **Important note:**
Using keys larger then 1024 bits results in slow responses over HTTPS and the converter webinterface can take excessive amounts of time to load.

When using a self-signed certificate you will need to accept a one time security warning in your web browser when opening the webinterface via HTTPS.

In the web browser Click **Advanced...** > **Accept the Risk and Continue**.



3.1. Turning off insecure HTTP.

Steps for turning of the insecure HTTP access:

1. Open the converter webinterface in a web browser.
   The default IP is 169.254.100.10
2. Open the **HTTP** settings in the left sidepanel.
3. Select **Configuration**.
4. Set **Port** to 0.
5. Click **Submit**.

Note:
The webinterface will become inaccessible through the insecure HTTP connection right after this operation. You will need to enter the https:// prefix in the address bar and open the webinterface again.


M-Bus converter Web Interface
JC Elektronika s.r.o.
Logout
Status
CLI
CPM
Diagnostics
DNS
Email
Filesystem
FTP
Host
HTTP
IP Address Filter
Line
LPD
Modbus
Network
PPP
Protocol Stack
Query Port
RSS
SNMP
SSH
SSL
Syslog
System
Terminal
TFTP
Tunnel
XML
Statistics
Configuration
Authentication
HTTP Configuration
State: Enabled Disabled
Port: 0
Secure Port: 443
Secure Protocols: TLS1.0 TLS1.1 TLS1.2
Max Timeout: 10 seconds
Max Bytes: 40960
Logging State: Enabled Disabled
Max Log Entries: 50
Log Format: %h %t "%r" %s %B "%{Referer}i" "%{User-Agent
Authentication Timeout: 30 minutes
Help
This page displays and changes the current HTTP Configuration settings.
Copyright © JC Elektronika s.r.o. www.prevodniky.sk

4. Configuring SSL

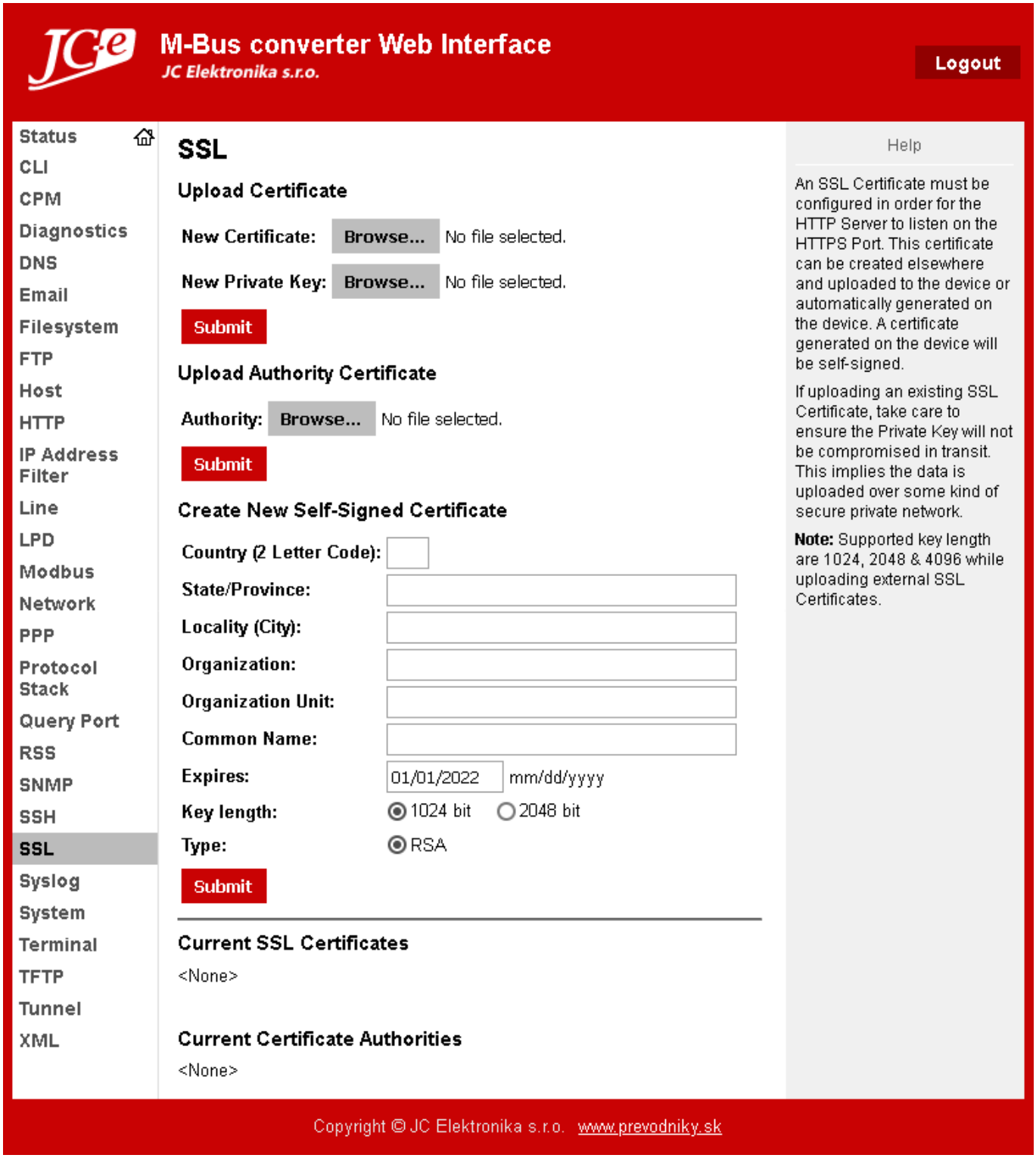
In order to use SSL tunneling for data transfers or HTTPS for accessing the webinterface an SSL certificate and a private encryption key needs to be configured. The converter can internally generate a certificate and a key or external ones can be uploaded to the converter.

Steps for creating an internally generated certificate:

1. Open the converter webinterface in a web browser.
   The default IP is 169.254.100.10
2. Open the **SSL** settings in the left sidepanel.
3. In the **Create New Self-Signed Certificate** fill in all the required details:
   Country (2 Letter Code), State/Province, Organization, Organization Unit, Common Name, Expires, Key length
4. Click **Submit** in the **Create New Self-Signed Certificate** section.

Steps for uploading your own certificate:

1. Create a private encryption key and certificate, see chapter: Creating a private key and a self-signed certificate.
2. Open the converter webinterface in a web browser.
   The default IP is 169.254.100.10
3. Open the **SSL** settings in the left sidepanel.
4. In the **New Certificate** setting click the **Browse...** button and select the certificate file (`cert_converter.pem`).
5. In the **New Private Key** setting click the **Browse...** button and select the key file (`key_pkcs8_converter.pem`).
6. Click **Submit** in the **Upload Certificate** section.


M-Bus converter Web Interface
JC Elektronika s.r.o.
Logout
Status
CLI
CPM
Diagnostics
DNS
Email
Filesystem
FTP
Host
HTTP
IP Address Filter
Line
LPD
Modbus
Network
PPP
Protocol Stack
Query Port
RSS
SNMP
SSH
SSL
Syslog
System
Terminal
TFTP
Tunnel
XML
SSL
Upload Certificate
New Certificate: Browse... No file selected.
New Private Key: Browse... No file selected.
Submit
Upload Authority Certificate
Authority: Browse... No file selected.
Submit
Create New Self-Signed Certificate
Country (2 Letter Code):
State/Province:
Locality (City):
Organization:
Organization Unit:
Common Name:
Expires: 01/01/2022 mm/dd/yyyy
Key length: 1024 bit 2048 bit
Type: RSA
Submit
Current SSL Certificates
<None>
Current Certificate Authorities
<None>
Help
An SSL Certificate must be configured in order for the HTTP Server to listen on the HTTPS Port. This certificate can be created elsewhere and uploaded to the device or automatically generated on the device. A certificate generated on the device will be self-signed.
If uploading an existing SSL Certificate, take care to ensure the Private Key will not be compromised in transit. This implies the data is uploaded over some kind of secure private network.
Note: Supported key length are 1024, 2048 & 4096 while uploading external SSL Certificates.
Copyright © JC Elektronika s.r.o. www.prevodniky.sk

5. Creating a private key and a self-signed certificate

To create private encryption key and a self-signed certificate we will be using the openssl command line tool.

Openssl comes bundled with the stunnel application, there should be no need to install it if stunnel is installed.
It can be downloaded as a standalone tool at: https://openssl-library.org/source/

Documentation for this tool can be found here:
https://wiki.openssl.org/index.php/Command_Line_Utilities

The converter needs a the key to be stored in a format called PKCS#8. Openssl has an internal conversion tool that can convert the key to this format. More details can be found here:
https://docs.openssl.org/master/man1/openssl-pkcs8/

Note:
A separate certificate and key should be generated for the stunnel application and the converter.
Following the steps below individual files for the two sides should be generated:
`cert_stunnel.pem`, `key_stunnel.pem` for the stunnel
`cert_converter.pem`, `key_pkcs8_converter.pem` for the converter

Steps for creating a private key and a certificate:

1. Open command prompt as administrator.
   From the start menu find **Command Prompt** right click and choose **Run as administrator**.

   Note:
   This is only necessary if you installed stunnel in C:\Program Files (x86)\stunnel as this location requires administrative privileges to create or change files.

2. Change directory to the openssl location with this command:

```
cd C:\Program Files (x86)\stunnel\bin
```

3. Create a key and certificate using openssl by entering the following command:

```
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -sha256 -days 365
```

   Note:
   The converter supports 1024, 2048 and 4096 key lengths which can be set with the `rsa:` parameter.
   The stunnel application requires a key that is at least 2048 bits long.

4. Enter the certificate details, if the fields are left empty default values will be used (shown in brackets):

```
Country Name (2 letters code):
State or Province (full name):
Locality Name (eg, city):
Organization Name (eg, company):
Organizational Unit Name (eg, section):
Common Name (eg, server FQDN or YOUR name):
Email Address:
```

5. In order for the key to be used in the converter it needs to be converted into the pkcs8 format.
   This command will convert `key.pem` and save it into a new file named `key_pkcs8.pem`.

```
openssl pkcs8 -in key.pem -nocrypt -traditional -out key_pkcs8.pem
```

   Note:
   The key format does not need to be changed when it is used with stunnel.

6. Rename the files according to their destination device.
   This can be done in Windows Explorer or by using the following commands:

   For converter

```
ren cert.pem cert_converter.pem
ren key_pkcs8.pem key_pkcs8_converter.pem
```

   For stunnel, a different certificate and key should to be generated.
   Repeat the process from step 3. Then rename the newly created files.

```
ren cert.pem cert_stunnel.pem
ren key.pem key_stunnel.pem
```

7. To use the key and certificate in the stunnel application they will have to be moved
   from `C:\Program Files (x86)\stunnel\bin`
   to `C:\Program Files (x86)\stunnel\config`

   This can be done in Windows Explorer or by using the following commands:

```
move /Y cert_stunnel.pem "C:\Program Files (x86)\stunnel\config"
move /Y key_stunnel.pem "C:\Program Files (x86)\stunnel\config"
```

8. If you will want to use certificate validation, copy the converter certificate to the `config` folder with a new name `ca_cert_converter.pem`

```
copy /Y cert_converter.pem "C:\Program Files (x86)\stunnel\config\ca_cert_converter.pem"
```