Encrypted data with Digi One IAP

February 2018

90000646

Contents

1 Introduction 3

1.1 Overview 3

2 Using SLL v3 or TLS v1 4

2.1 Digi products supporting SSL/TLS 4

2.2 Issues/caveats 4

2.3 Partial application example: 5

3 Using encrypted RealPort 5

3.1 Digi RealPort drivers supporting SSL/TLS 5

3.2 Enabling encryption 6

3.3 Issues/caveats 7

4 Using Secure Shell (SSH or OpenSSH) 8

4.1 Using PuTTY for SSH access to serial port 8

4.2 Using OpenSSH for SSH access to serial port 8

5 Troubleshooting and FAQ 8

5.1 Why is my connection rejected? 8

5.2 My connection always times out 8

1 Introduction

Abstract

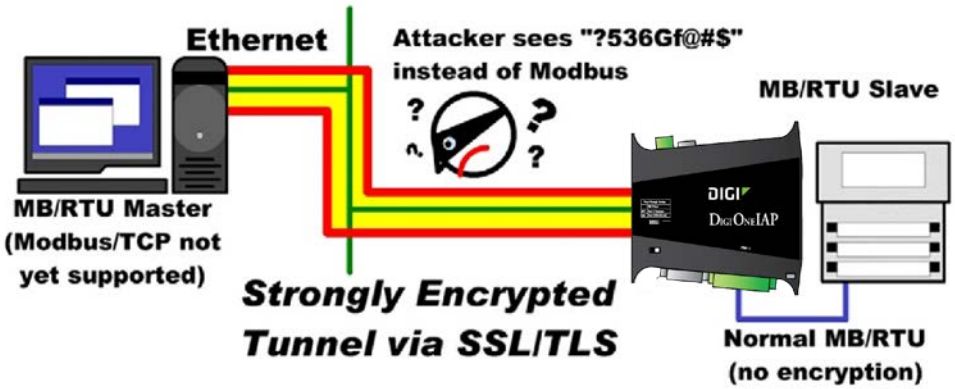
The Digi One IAP supports strong 128-bit AES encryption, SSL/TLS, and SSH. This enables privacy for your automation data, and can be critical if your common Modbus or Rockwell traffic is passing through a firewall into semi-public networks.

1.1 Overview

Security is an important topic. Books and tutorials exist for all levels of interest, from users who just want to know the subject to the byte-level protocols for programmers.

The Digi One IAP features the following:

- **Strong encryption for extreme privacy**. Anyone sniffing the network will not see or be able to decode the IA protocol packets.
- **Complete integrity**. A 160-bit SHA1 message digest is vastly safer than 8-, 16-, or 32-bit CRC. In human terms this means you'll never see corrupted data in the secure tunnel arrive with error undetected.



The TCP/IP stack in the Digi One IAP supports SSL v3.0 and TLS v1.0 (also called SSL v3.1). Its preferred ciphers are the US government-promoted 128-bit or 256-bit AES ciphers. It also supports the older 3DES cipher, but this should not be used if AES is available, since it is much more CPU intensive.

2 Using SLL v3 or TLS v1

Your Digi One IAP firmware must be at release E or above. There is nothing to configure within the Digi One IAP. Open an SSL/TLS session to TCP port 2601 for the first port, and TCP port 2602 for the second port. Then you can send Modbus or other supported protocols. This applies to other Digi products, for example a 16-port TS16 will have TCP sockets 2601 to 2616 waiting for SSL/TLS clients.

2.1 Digi products supporting SSL/TLS

While this application note targets Digi One IAP users, the same information can be used for any of the Digi products listed below.

The following products with firmware 82000747_W1, released February 19, 2016:

- Digi One TS (including W and H models)
- Portserver TS 2/4 MEI
- Portserver TS 2/4 W
- Portserver TS 2/4 H
- Portserver TS 1/3 + Modem

The following products with firmware 82000684_W1, February 19, 2016:

- Portserver TS 8/16
- Portserver TS 8/16 MEI

The following product with firmware 82000770_W1, released February 19, 2016:

- Digi One IAP (previously called Digi One IA RealPort)

2.2 Issues/caveats

- The Digi One IAP will not negotiate a session using SSL v2.0. Make sure your client is configured to initiate using SSL v3.0 or TLS v1.0.
- Due to the intensive public key calculations required, the open process can take 30 seconds or more. After this, data flows TCP socket-like without noticeable delay or added latency. This is because data transmission is done using temporary AES keys and not public keys.
- To obtain the best performance, configure your client to prefer the AES cipher. It is considered excellent strong-encryption by the US government and was designed to reduce CPU load without sacrificing security strength.
- The protocol encapsulated in SSL/TLS must match the serial protocol on the corresponding serial port. In other words, if a Modbus/RTU slave is on serial port 1 and an Allen-Bradley DF1 slave is on serial port 2, you must send encrypted Modbus/RTU to TCP port 2601 and encrypted AB/DF1 to TCP port 2602.

- Because SSL/TLS requires data to be moved in fixed-size blocks, there is a slightly increased risk that data “streams” will have gaps between fragments. The IAP firmware in multi-master mode corrects this, but it may show up when the Digi One IAP is in a non-IA mode.

2.3 Partial application example:

There are too many possible tools to show all of them, but here is a simple Python example using the tlslite package to send a Modbus/RTU packet by SSL/TLS to the Digi One IAP. Network programmers will recognize that use of SSL is not much more complex than TCP sockets. Only the start-up process is different and affects timing.

You can download Python from www.python.org and tlslite from http://trevp.net/tlslite.

```
… (other code not shown)
import socket
from tlslite.api import *
…
# open a normal socket to TCP port 2601, increase timeout to 30 seconds mySock = socket.socket(
socket.AF_INET, socket.SOCK_STREAM) mySock.settimeout( 30.0)
mySock.connect( “192.168.1.20”, 2601)

# activate SSL/TLS within this socket – forcing minimum SSL3.0/TLS,
# suggesting AES, and not requiring authentication sslSession = TLSConnection( self.sock)
sslSets = HandshakeSettings() sslSets.cipherNames = [ "aes128", "aes256"] sslSets.minVersion = (3,0)
sslSession.handshakeClientCert()

…
# after opened successful, reduce timeout to 2 seconds mySock.settimeout( 2.0)

# send Modbus/RTU request sslSession.write( modbusRequest)

# receive Modbus/RTU response modbusResponse = sslSession.read( 256)

sslSession.close() mySock.close()
```

3 Using encrypted RealPort

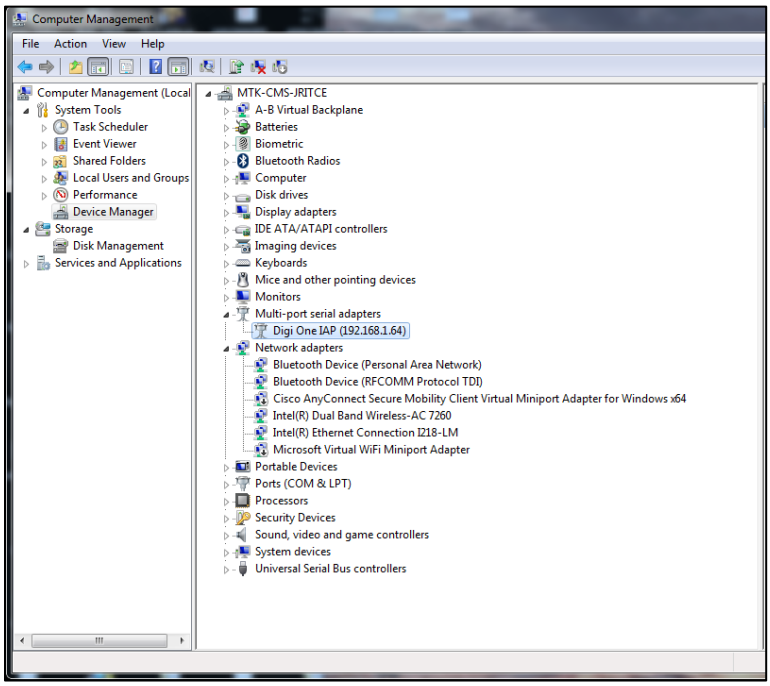
3.1 Digi RealPort drivers supporting SSL/TLS

- RedHat Enterprise Linux 2.1, 3, 4, 5.x, 6.x, 7.x
- RedHat 7.2, 7.3, 8, 9
- Suse 8.1, 8.2, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, 10.2, 11.1, 11.2
- Fedora Core 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17
- Mandriva Linux 8.2, 9.0, 9.1, 9.2, 10.0, 10.1, 2006, 2007.1, 2008
- Debian 3.0.rl, 3.1, 4, 6.0
- Ubuntu 7.04, 7.10, 8.04, 8.10, 12.04
- Linux Kernels support 2.4.x, 2.6.x,3.x.x,4.x.x (Up and SMP)
- Windows XP, Server 2003, 2008, 2008 R2 X86/X64, Windows 7, Windows 8 & 8.1, Server 2012, 2012 R2, Windows 10, Server 2016

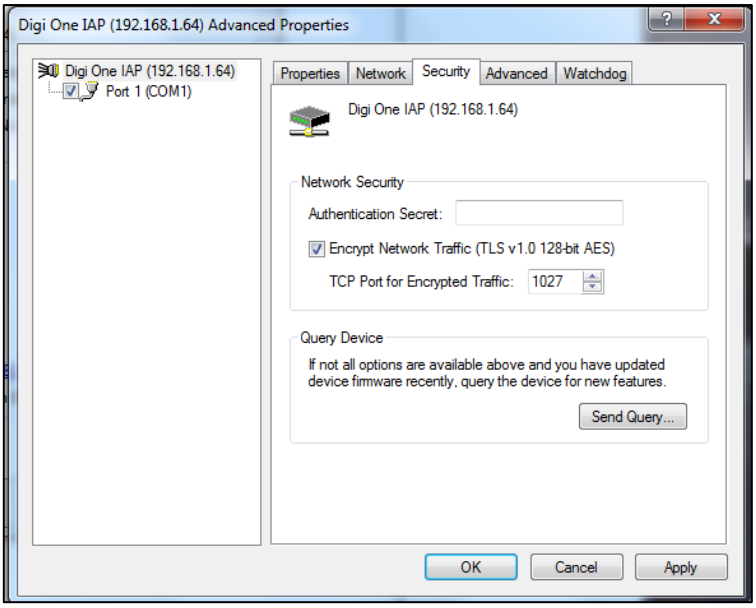
- SCO Openserver - Supports v5.0.4c, 5.0.5, 5.0.6, 5.0.7
- Solaris - Supports v2.6, 2.7, 2.8, 2.9 for SPARC and Intel

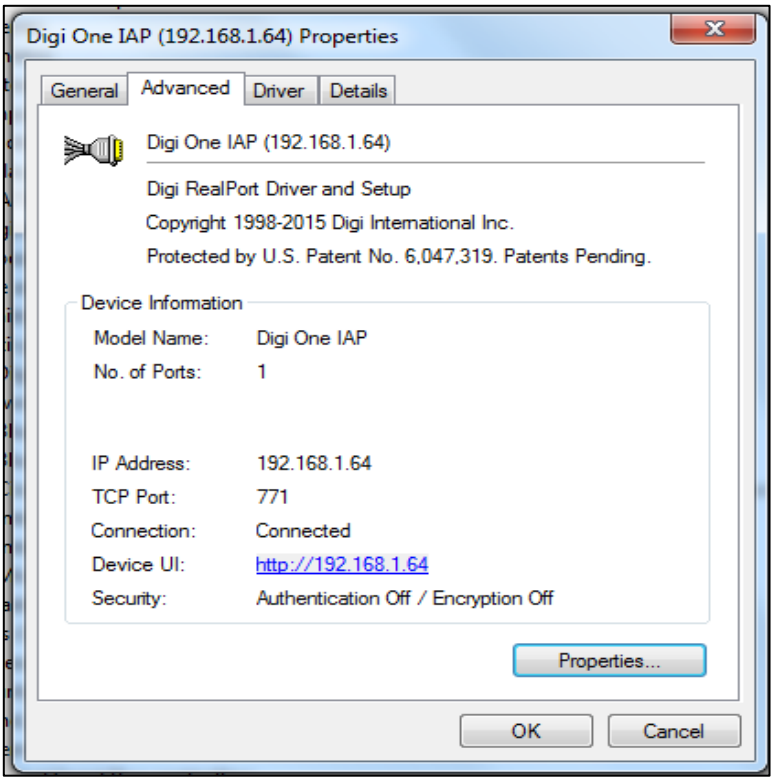
3.2 Enabling encryption

1. From the device manager, select the Digi One IAP device to enable encryption.
2. From the **Properties** dialog **Advanced** tab, select the next **Properties** button.



3. On the **Network** tab, enable **Encrypt Network Traffic**. In this example, COM4 and COM5 will be encrypted with TLS v1.0 and 128-bit AES.





3.3 Issues/caveats

The only real impact on your serial application may be a change in timeout behavior. Digi RealPort usually opens the SSL/TLS session independent of your application, but during network interruptions and recovery these could affect your application.

4 Using Secure Shell (SSH or OpenSSH)

An alternative to SSL/TLS is Secure Shell version 2 or SSH2. The core difference is that while SSL/TLS is an encrypted pipe to be used by any client/server application, SSH is a specific client/server application pair using encryption. You can find out more information at https://www.openssh.org/. Although it has been expanded to include FTP and other services, SSH was really designed to be a secure encrypted telnet for remote access.

Two strong advantages for using SSH with current Digi firmware are:

1. We support authentication by standard SSH login, so you can securely restrict and limit remote client access.
2. We support outgoing (Digi-as-client) SSH connections so a serial device can initiate the secure connection.

4.1 Using PuTTY for SSH access to serial port

An open SSH2 client that you can use to securely access a remote serial device is PuTTY available from www.chiark.greenend.org.uk/~sgtatham/putty/latest.html. PuTTY offers clients for Windows, UNIX, Linux, and other common operating systems.

4.2 Using OpenSSH for SSH access to serial port

To enable forwarding from TCP port 2101 on the local host to the serial port of the Digi DS/TS using OpenSSH use:

```
ssh -L 2101:localhost:2501 -l <user> -p 2501 <ip address of DS/TS>
```

5 Troubleshooting and FAQ

5.1 Why is my connection rejected?

SSL v2 is considered very insecure and is not supported by Digi. The Digi One IAP will not negotiate a session using SSL v2.0. Therefore, make sure your client application attempts to initiate the SSL session using SSL v3.0 or TLS v1.0 (also called SSL v3.1).

5.2 My connection always times out

SSL/TLS is using Public Key techniques to open the session. This means complex math exists to encrypt data in one direction, but that—even with today's high-end computers—it will take over 100 years to reverse this by brute force. This is the meaning of "strong encryption".

The 1024-bit Diffie-Hellman algorithm used to negotiate an SSL/TLS session can take the Digi One IAP up to 30 seconds to complete. Thus, your client must be willing to wait at least 30 seconds for the initial encrypted socket to open.

After this initial open you won't notice any extra delay. Protocol transactions can proceed at normal speed and with normal latencies.