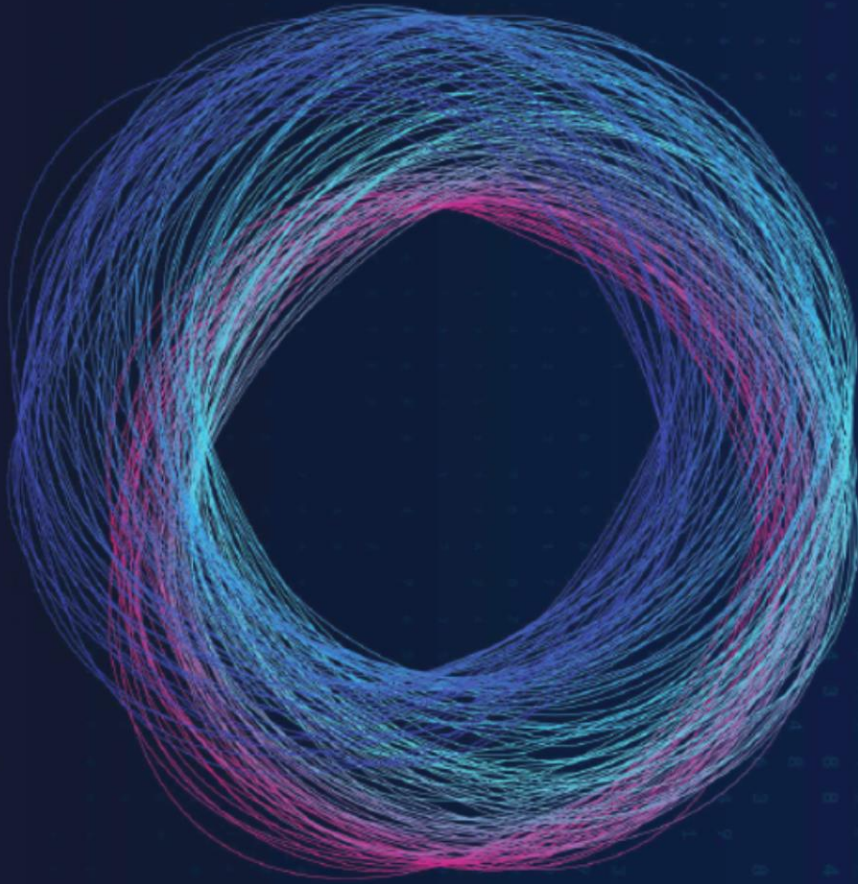


THALES


QUANTUMXCHANGE



Document Version

Phio TX™ Version	Date Published
1.3.0	May 2020

Trademarks, Copyrights

© 2020 Quantum Xchange and Thales. All rights reserved. Quantum Xchange and the Quantum Xchange logo are trademarks and service marks of Quantum Xchange Inc. and/or its subsidiaries and are registered in certain countries. Thales and the Thales logo are trademarks and service marks of Thales DIS CPL USA, Inc. and/or its subsidiaries and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the property of their respective owners.

Disclaimer

All information herein is either public information or is the property of and owned solely by Thales DIS CPL USA, Inc. (Thales") and/or its affiliates who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information. Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Thales' information. This document can be used for informational, non-commercial, internal and personal use only provided that:

- ☐ The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- ☐ This document shall not be posted on any network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities. The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Thales reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time. Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and noninfringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result

from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

Table of Contents

Document Version 2

Trademarks, Copyrights..... 2

Disclaimer 2

Preface 5

 Audience5

 Document Conventions.....5

 Support Contact.....5

Introduction 6

 About Phio Trusted Xchange (TX)6

 Architecture.....6

 Prerequisites6

Integrating Phio TX with Thales 7

 Configure Phio TX nodes7

 Define Clients (Encryptors)7

 Define ETSI Server Parameters8

 Configuring EQKD on the Encryptor10

Test Integration 14

Preface

Audience

This Integration Guide is intended for administrators tasked with integrating PhioTX with Thales High Speed Encryptors.

All products manufactured and distributed by Quantum Xchange and Thales are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

Document Conventions

This document employs **examples**, **code snippets**, and **notes**:

Examples

Examples are provided to illustrate different scenarios administrators may encounter.

Example 1: Scenario description

Code Snippets

Code snippets are intended to be copied exactly as written or modified as described in the text.

Code Snippet

Notes

Notes provide additional or contextual information.

Note: Notes provide additional or contextual information that may not be immediately obvious, but that isn't essential to performing the integration.

Support Contact

If you have additional questions or require further support, please reach out to Quantum Xchange at support@quantumxc.com.

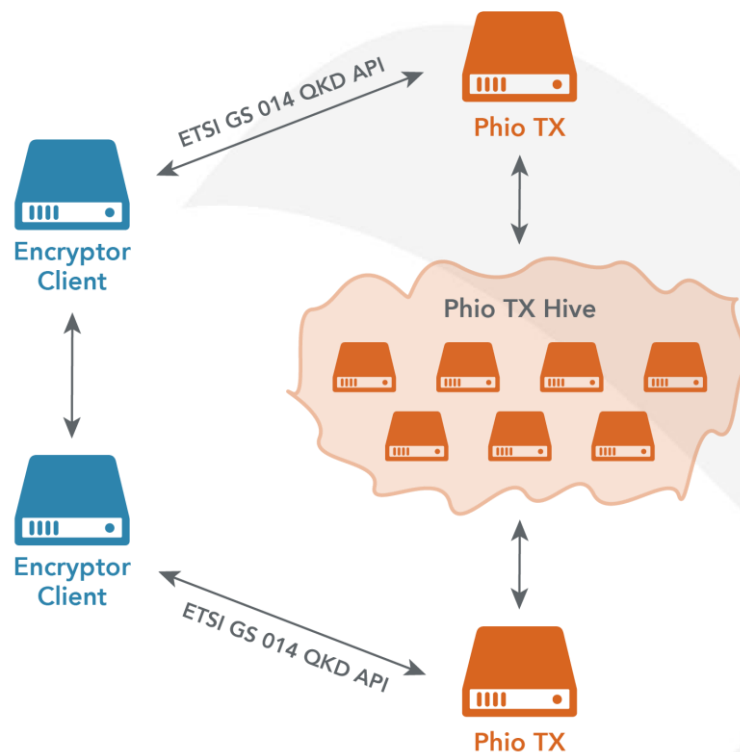
Introduction

About Phio Trusted Xchange (TX)

Phio Trusted Xchange™ (TX) provides secure, out-of-band key distribution to encryptors or other security entities, enabling quantum safe encryption. With Phio TX, an attacker cannot steal keys by tapping a network connection that is secured with public key encryption (PKE), because they will not have all the keys required to decrypt the traffic. This protects existing PKE tunnels, such as those used to exchange keys between encryptors, from attacks from quantum computers, or any number of attacks against PKE.

Architecture

Each encryptor needs to be able to communicate to a Phio TX node and receive keys via a simple REST API that uses the ETSI QKD (ETSI QKD GS 014) protocol. An encryptor is called a Client of the Phio TX node. Phio TX nodes that talk directly to another node are called Peers of each other. The Phio TX nodes need to be able to pass keys to each other, but they do not necessarily need to be peers. They need only to be able to pass keys via one or more intermediate peer Phio TX nodes. (For more information, see the Phio TX Install and Admin Guide.)



Prerequisites

Before you proceed with the integration, complete the following:

- PhioTX configuration as described in the PhioTX Install and Admin Guide
- Thales encryptor configuration

Integrating Phio TX with Thales

For the purposes of this document, we will assume that there are two Thales encryptors – Encryptor 1 and Encryptor 2. Each encryptor gets keys from its own Phio TX node, which may or may not be a direct peer of the node that has the other encryptor as its client, as shown in the architecture diagram above.

Configure Phio TX nodes

Configure the Phio TX host as described in the Phio TX Installation and Admin Guide. This host will communicate with Encryptor 1 as its client and allow the client to communicate with the rest of the Phio TX network. The Phio TX host must be able to send keys to the Phio TX node that has Encryptor 2 as its client.

Define Clients (Encryptors)

Edit the “clients” parameter in the Phio TX configuration file (the default is *tx_conf.yaml*) to define encryptors or other devices that will request keys from the Phio TX. Clients may be defined by hostname/FQDN, IP address, or both, per the following examples:

Example 1: Hostname/FQDN only

In this scenario, the name must be resolvable by DNS. Client certificate validation is then also supported, so long as the CN in the certificate presented by the client matches the configured value.

```
clients:
  - name: enc1.qxc-customer.net
```

Example 2: Hostname and IP address.

In this scenario, Phio TX will create a static mapping for that hostname in the *hosts* file. Client certificate validation is supported.

```
clients:
  - name: enc2
    addr: 10.80.12.4
```

Example 3: IP address only.

Client certificate validation is not supported. (This is the setting required for encryptors where the QKD configuration uses only an IP address to identify the encryptor.)

```
clients:
  - addr: 10.80.12.5
```

Define ETSI Server Parameters

Clients (encryptors) retrieve keys from the ETSI Server component of Phio TX.

1. Modify the default configuration for the ETSI Server, use *vi* to set the following parameters:

Parameter	Possible values	Notes
etsi_no_client_cert	yes, no (default)	Changing this to “yes” will ignore client certificate validation for all connection attempts to the ETSI server.
etsi_tls_version	0, 1, 2, -1 (default)	This corresponds to the TLS 1.x version used by the client when connecting to the ETSI Server. For example, changing the setting to “0” will allow TLS 1.0, which is required for certain encryptors. “-1” will accept all TLS versions. The default is “-1”
etsi_port	0-65535 (default: 443)	The TCP port on which the ETSI Server listens for connections

2. Save the Configuration File and exit *vi*.

3. Apply the Phio TX Configuration:

Run the `tx_install_cf` command to test the configuration file: (The example below shows the configuration file is `tx_conf.yaml`)

```
tx_install_cf tx_conf.yaml
```

Fix any errors noted in the output of the command. Once you are ready to finalize the configuration, add the `-y` switch to apply it.

```
tx_install_cf -y tx_conf.yaml
```

4. Create Certificates

Follow the procedure below to apply certificates to the Phio TX appliance. Phio TX runs two REST servers:

- The “TX Server” used for communication between Phio TX nodes. This server is only active when the Phio TX has configured peers.
- The “ETSI Server,” used for key retrieval by clients (encryptors). The ETSI server is only active when the current Phio TX serves encryptor(s), rather than being an intermediate key delivery node.

Both servers require signed certificates to operate. The procedure for applying the certificates to the respective REST servers is very similar.

Note: Certificates are only required for the REST servers configured to run on this Phio TX. For example, there's no need for ETSI certificates on Phio TX node that only operates as a load-balancing or a media-changing pass-through node.

Note: The following steps will generate Certificate Signing Requests (CSRs) based on the current hostname, so before proceeding, run the *hostname* command to verify that the hostname is set to the desired value.

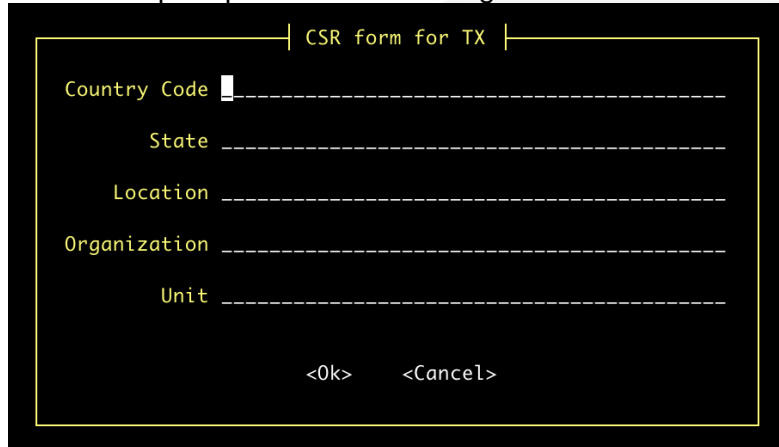
Note: The CSRs for both the TX Server and the ETSI Server are currently based on RSA 4096-bit private keys. Some encryptor vendors may require different algorithms and/or key sizes for interoperability. In this situation, you must first generate a private key with the needed parameters, import the private key into the ETSI Server configuration, and then generate the CSR. For example, to generate and import a private key based on the ECC secp521r1 curve, use the following commands. Then proceed to *Step a)*:

```
openssl ecparam -genkey -name secp521r1 -out private-ec.key  
tx_install_etsi_private_key private-ec.key
```

- a) Generate an ETSI certificate signing request (CSR) with the following command:

```
tx_generate_etsi_csr
```

You will be prompted for the following details:



The screenshot shows a terminal window with a yellow border. At the top, it says "CSR form for TX". Below this, there are five fields with labels and dashed lines for input: "Country Code", "State", "Location", "Organization", and "Unit". At the bottom of the form, there are two options: "<Ok>" and "<Cancel>".

Enter the required information, and a CSR file will be written to the current directory.

- b) Copy the CSR to an external system and using your existing Public Key Infrastructure (PKI), generate an X509v3 certificate based on the CSR.
- c) Copy the newly created certificate, as well as the corresponding root CA certificate(s) to the Phio TX appliance.
- d) Run the following commands to import the certificates:

For ETSI Server:

```
tx_install_etsi_ca [filename_of_tx_etsi_cert.pem]***  
tx_install_etsi_cert [filename_of_etsi_cert.crt]
```

*****Note:** This step is required even if using the same CA certificate that was used for the TX Server.

Once the certificates are successfully imported, the respective services will automatically restart.

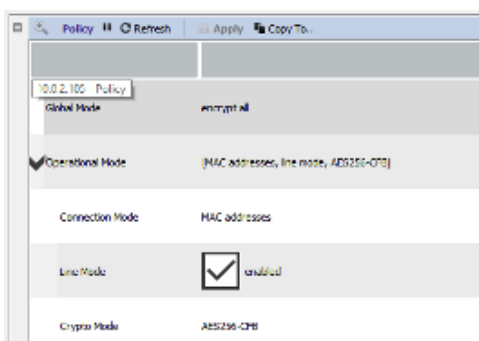
Configuring EQKD on the Encryptor

These examples use the CM management software to configure the encryptors. If you are managing the encryptors using another tool, the basic outline of steps should remain the same, even though the screenshots won't match.

At a high level the steps required are as follows:

- Configure encryptors first in Line Mode and ensure encrypted tunnels are up and running
- Enable EQKD on each encryptor
- Create CSRs for the certificates used for the ETSI connection
- Generate the ETSI certificates on an external CA (the same one used to create the Phio TX ETSI Certificates)
- Import the PEM file for the CA that generated the ETSI certificates into each encryptor
- Import the PEM file for the ETSI certificate generated from the CSR into each encryptor
- Check to see if QKD keys are being delivered

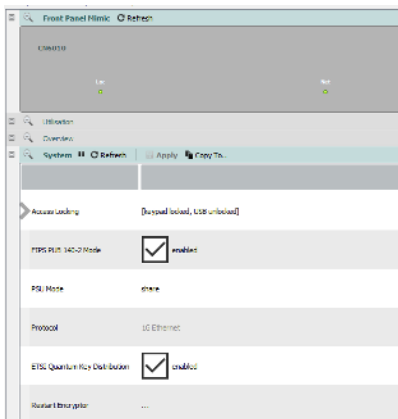
First, make sure the encryptors are in Line Mode and using AES256-CFB encryption by checking the Policy pane under the CM console Manage page:



Note: Encryptors rotate keys at a configurable interval. This rotation determines the frequency of QKD key requests. To speed up testing and verification that your configuration is correct, we recommend that you set the Key Update Interval in the Connections pane to 1 minute until the configuration is verified as running correctly. This allows you to see the key request occurring more quickly, which greatly speeds up troubleshooting. You can change this back to the original value (default is 60 minutes) when you are finished.

Connections	
Local Encryptor Mac Address	nn
Key Update Interval(min)	00:D0
Dead Peer Detection Interval(sec)	0
Authentication Interval(hours)	0

Next, enable EQKD on each encryptor by checking the ETSI Quantum Key Distribution enabled box under the System pane of the Management page on the CM console:



First Panel Home Refresh

System Refresh Apply Copy To...

Access Loading [Access loaded, USB unlocked]

PPS Plus 140-2 Mode ☒ enabled

PSI Mode none

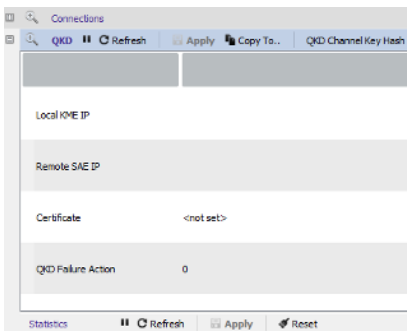
Protocol 10 Ethernet

ETSI Quantum Key Distribution ☒ enabled

Monitor Interruptor ...

After you apply the changes to the EQKD settings you will need to restart the encryptors for the change to take effect. Note that while this should happen automatically, it may require a manual restart.

Once the system has restarted, you can see the QKD panel on the Management page of the encryptor. generate the ETSI certificate requests. All the entries should be blank:



Connections

QKD Refresh Apply Copy To... QKD Channel Key Hash

Local KME IP

Remote SAE IP

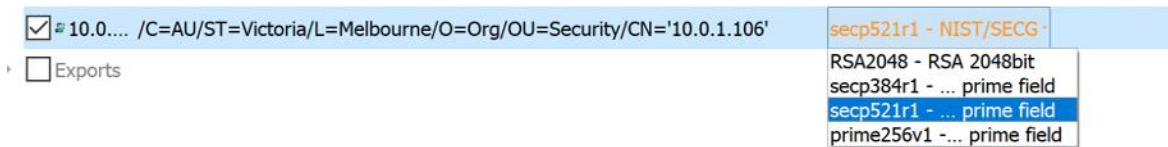
Certificate <not set>

QKD Failure Action 0

Statistics Refresh Apply Reset

Before you can configure the QKD settings, you first need to generate ETSI certificates that will be trusted by the Phio TX host. On the Certify page of the CM console, select the encryptors for which you will be creating ETSI certificates by checking the box next to their name in the list. (You can select more than one encryptor at this step and unique CSRS will be created one by one.)

You also need to select the proper certificate type for your environment. In this case, we are creating a CSR for a certificate using the secp521r1 curve:



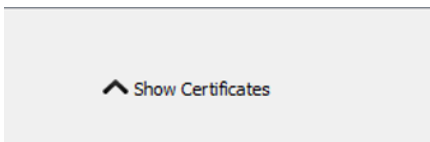
Next, click the External CA tab on the right side of the page and then click the Save CSRs to File button at the bottom of the page:



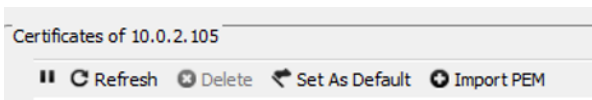
You will be prompted for a file name and location to save each CSR that is generated, one after the other.

Take those CSRs and generate a standard TLS certificate using the same CA that you used to generate the ETSI certificates for the Phio TX Host.

Once you have the certificate files (in CRT or PEM format) ensure that the file extension is .PEM and then import the certificates by going to the Certify page on the CM console, selecting the encryptor (only one at a time) that you want to import the ETSI cert to, and then click the Show Certificates button at the bottom of the screen:



You will get a new pane with an Import PEM button, which allows you to Import the certificate you have created.

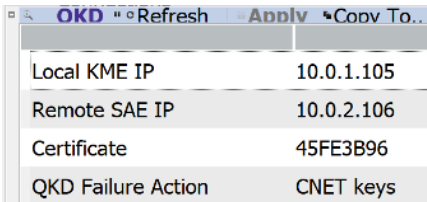


Once the certificates are installed, use the Management page to view the Certificates panel for each encryptor, as shown below. You may need to adjust the display width if each column to see all the information clearly:

Certificates Refresh Delete Set As Default Import PEM								
ID	Type	Identifier	PK Algorithm	PK Size	Days Left	Status	Usage	Signed By
3	X509 EN	21D8D47C	RSA	2048	1782	signed	in use	8: 80713F4D
4	X509 EN	45FE3B96	EC	521	352	signed	not in use	10: FC30F29A
5	X509 EN	A8B91165	EC	521	1813	signed	not in use	8: 80713F4D
6	X509 EN	C3E63B12	RSA	2048	352	signed	not in use	9: B70E131D
7	X509 EN	FD8CF3CB	EC	521	1813	signed	not in use	8: 80713F4D
8	X509 CA	80713F4D	RSA	2048	5269	signed	in use	self-signed
9	X509 CA	B70E131D	RSA	2048	1615	signed	not in use	self-signed
10	X509 CA	FC30F29A	EC	521	3559	signed	not in use	self-signed

Note: Each certificate has a unique ID, and that is required to use that certificate for EQKD. The certificate you imported for ETSI will have been signed by a CA, and will NOT be the self-signed certificate. Take note of the Identifier for this certificate as you need to select this certificate in steps below.

Next, configure the EQKD settings as shown below:



Local KME IP	10.0.1.105
Remote SAE IP	10.0.2.106
Certificate	45FE3B96
QKD Failure Action	CNET keys

- The Local KME IP is the IP address of the Phio TX host providing keys to THIS encryptor
- Remote SAE IP is the IP address of the other encryptor in this pair that will also use QKD
- The Certificate is a drop-down list of all certificates install. Select the certificate ID for the ETSI certificate you imported earlier into this encryptor.
- QKD Failure Option determines what happens when keys cannot be provided from the Phio TX host. The default is CNET Keys which will cause the encryptor to fall back to using standard TLS until a new out-of-band key can be retrieved from Phio TX.

Click Apply and then verify that tunnels are up. If QKD keys cannot be retrieved, you will see a message “Unable to Receive QKD keys...” on the front panel of the encryptor.

Test Integration

Test the configuration by running the following commands on the Phio TX host:\

```
tx_status
```

You should see the certificate information for all the certificates you have installed on the Phio TX node.

```
txh -l
```

You should see the Phio TX peers, as well as the Encryptor client appear, with green status Information.