TREK

CRYPTOGRAPHY SERVICES

TUTORIAL

January 2021

TABLE OF CONTENTS

PARAGRAPH PAGE

FIGURES

FIGURE PAGE

TABLES

TABLES PAGE

1 Welcome

The Telescience Resource Kit (TReK) is a suite of software applications and libraries that can be used to monitor and control assets in space or on the ground.

This tutorial describes TReK cryptography services. It describes how to use common cryptography functions available in the TReK software.

This tutorial uses screen dumps taken on a Windows computer. However, the steps are the same regardless of whether you are running on Windows or Linux.

2 Technical Support

If you are having trouble installing the TReK software or using any of the TReK software, please contact us for technical assistance:

TReK Help Desk E-Mail, Phone & Fax:

E-Mail: trek.help@nasa.gov
Telephone: 256-544-3521 (8:00 a.m. - 4:00 p.m. Central Time)
Fax: 256-544-9353

If you call the TReK Help Desk and you get a recording please leave a message and someone will return your call. E-mail is the preferred contact method for help. The e-mail message is automatically forwarded to the TReK developers and helps cut the response time. The HOSC Help Desk (256-544-5066) can provide assistance as needed and is available 24x7.

3 Introduction

Cryptography provides the means to protect data from unauthorized disclosure or modification during transmission or storage. TReK provides cryptography services that can be used to encrypt and decrypt files and data streams. These capabilities are available across various TReK software applications and libraries. TReK uses OpenSSL's FIPS 140-2 validated cryptographic module and public/private key pairs to encrypt and decrypt files and packets. TReK cryptography services are available on 32 bit and 64 bit Linux operating systems and 64 bit Windows operating systems. TReK cryptography services are not available on 32 bit Windows operating systems.

TReK cryptography services use symmetric encryption. Symmetric encryption uses cryptography keys along with an encryption algorithm to encrypt data so it cannot be distinguished from random garbage data. A cryptography key is just a long string of binary data that can be stored in a file or in memory. An analogy is a lock and a key. Cryptography keys along with an encryption algorithm “lock” the data. Modern

cryptography is based on the idea that the key you use to encrypt your data can be made public while the key that is used to decrypt your data can be kept private. TReK uses the Advanced Encryption Standard (AES) algorithm certified by the United States government.

Sometimes the easiest way to understand a new concept is by looking at an example. Cryptography is a complex topic. This is a very simplified example but sufficient if you just need to know enough to use TReK cryptography services.

Suppose Sulu and Chekov want to exchange some data, but they want to keep the data secret. They decide to use symmetric encryption. To use symmetric encryption, both Sulu and Chekov need to generate a Public Key and a Private Key. [Don't worry TReK has an application for that.] Once both Sulu and Chekov have generated their public and private key pair, they will need to pre-share their public key with each other (e.g. send it to each other via encrypted e-mail or use the Federation Large File Transfer which provides encrypted file transfers). Sulu will send Chekov his public key. Chekov will send Sulu his public key. Figure 1 shows the keys on each of their computers.

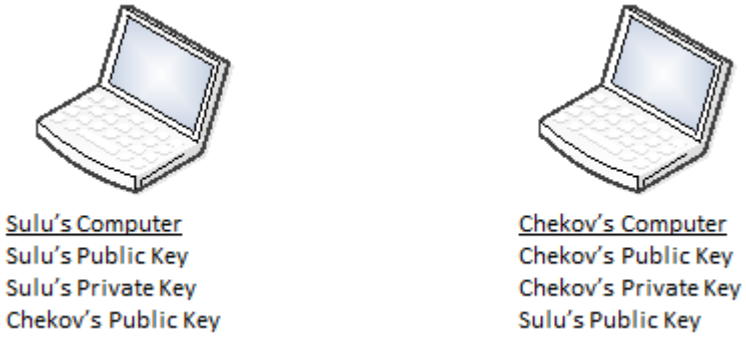

Figure 1 Cryptography Keys Example

When Sulu wants to send data to Chekov, he can use his own private key along with Chekov's public key to perform the encryption. [Behind the scenes, these two keys are used to generate a shared secret key along with a few more steps that make all this work.]. When Chekov receives the encrypted data from Sulu, only Chekov will be able to decrypt the message, since he has Sulu's public key and he has the one and only private key that goes with his public key. And vice versa if Chekov is sending data to Sulu.

When exchanging encrypted data with another party, you will need the other party's public key. When using TReK cryptography services, you will see GUI fields and API arguments to provide a Peer Public Key. The other party's public key is referred to as the "Peer" public key. You will need to have a copy of the other party's public key on your computer. TReK will need to know the absolute path to the location of the peer public key to perform encryption and decryption.

At a minimum when using TReK cryptography services you will need your public key, your private key, and the other party's public key (referred to as a peer public key).

It is possible to provide a passphrase for a private key to enhance security. If this is done the passphrase will be required in order to decrypt the data. The passphrase is used to wrap/encrypt the private key prior to storing it in the private key file. TReK uses a default passphrase when wrapping the key if the user does not provide one

So that's the basic information you need to know to use TReK cryptography services. Read on for details on how to generate keys and the various cryptography services that are available.

4 TReK Cryptography Services

This section describes how to configure the TReK software to perform various cryptography tasks. If you are new to these applications and libraries, please reference the associated user guides, tutorials, and on-line help for an introduction before reviewing the following material. The following material assumes you are familiar with these applications and only addresses tasks related to cryptography.

4.1 How to use TReK Crypt to Generate Cryptography Keys

TReK uses symmetric encryption. The TReK Crypt application provides the capability to generate public and private cryptography keys. Figure 2 shows the TReK Crypt application main window. To generate a public key and private key enter a cipher key name and cipher key directory for each key. If you would like additional protection when storing your private key in a file, you can also enter a passphrase. Once you have entered all the information push the Generate button to generate the keys. If you have two computers (such as a flight computer and a ground computer), you can use this application to generate a public and private key pair for each computer. To exchange encrypted data between the two computers, you will need to ensure each computer has a copy of the other's public key. The public key from another computer is a peer public key. TReK will use peer public key terminology when referring to another computer's public key.

Important: It is critical that you back up these keys and ensure they are never lost or stolen. Once you use these keys to encrypt data, there will be no way to decrypt the data without the associated keys. It is wise to make a copy of the keys and put them in a safe location prior to generating any more keys. If key files with the same name exist when you push the Generate button the application will ask you to confirm you want to overwrite, but it only takes one keystroke to accidently overwrite thereby losing the only copy of your keys.

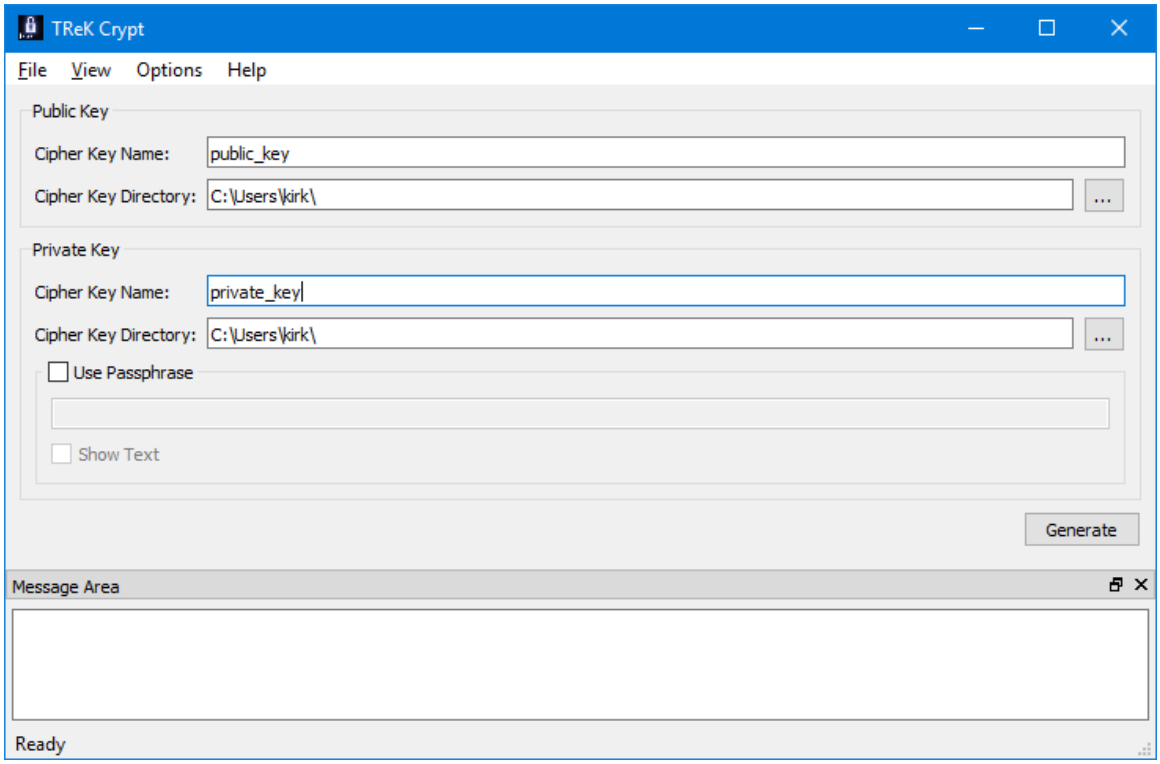

Figure 2 TReK Crypt Main Window

4.2 How to use TReK CFDP to Encrypt and Decrypt all CFDP Communication

The TReK CFDP application can be used to encrypt and decrypt all CFDP communication, making it possible to configure the CFDP application to encrypt and decrypt all CFDP transactions (e.g., "put", "get", "message", "create_file", "delete_file" ...). All packets are encrypted prior to transmission and decrypted upon receipt. This capability can only be used with Native CFDP. Figure 3 shows the CFDP Configure dialog with three remote entities in the list.

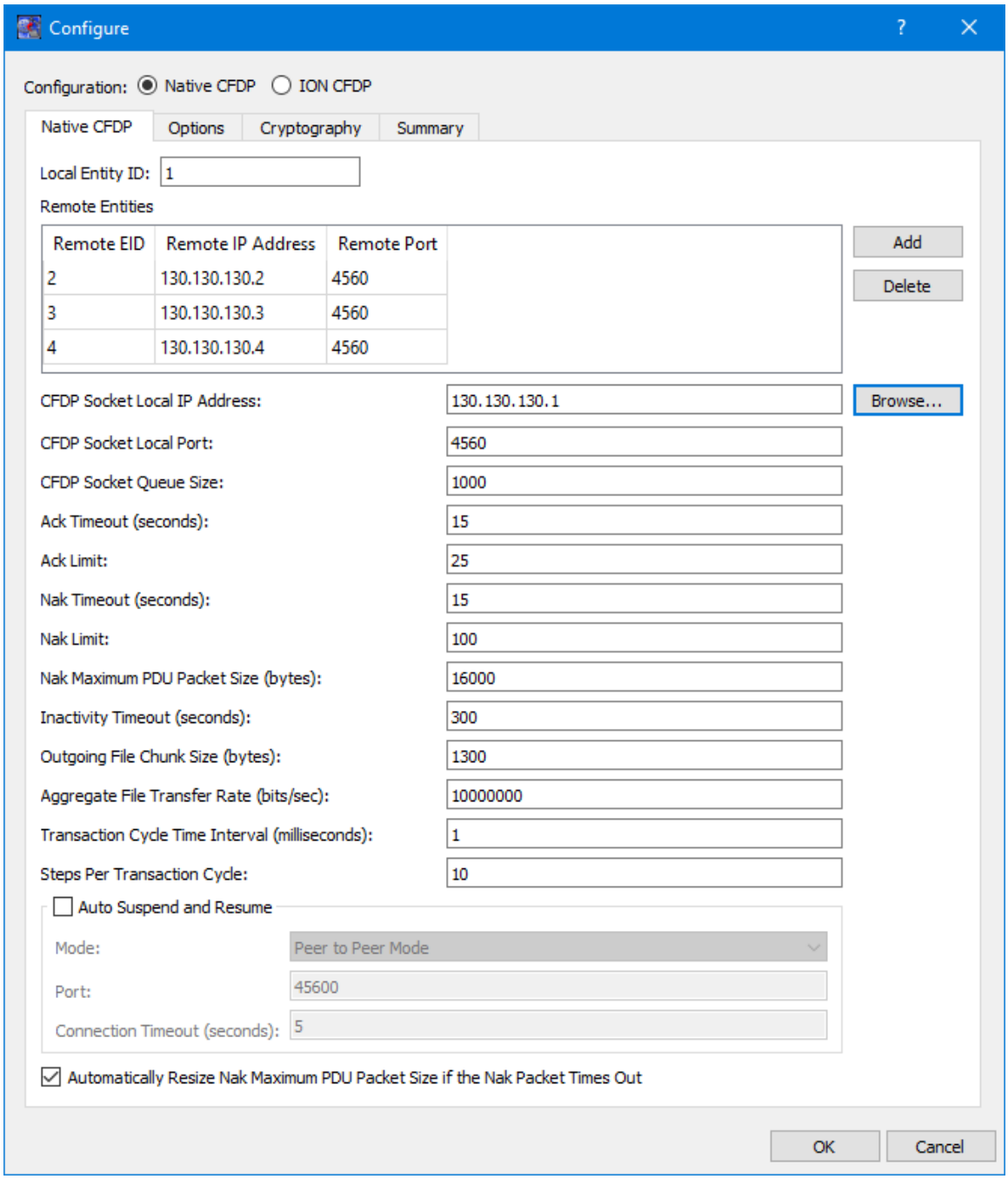

Figure 3 CFDP Configure Dialog (Native CFDP Tab)

Figure 4 shows the CFDP Configure Cryptography tab. To use CFDP cryptography capabilities you need to check the Cryptography checkbox and populate the fields in the General area.

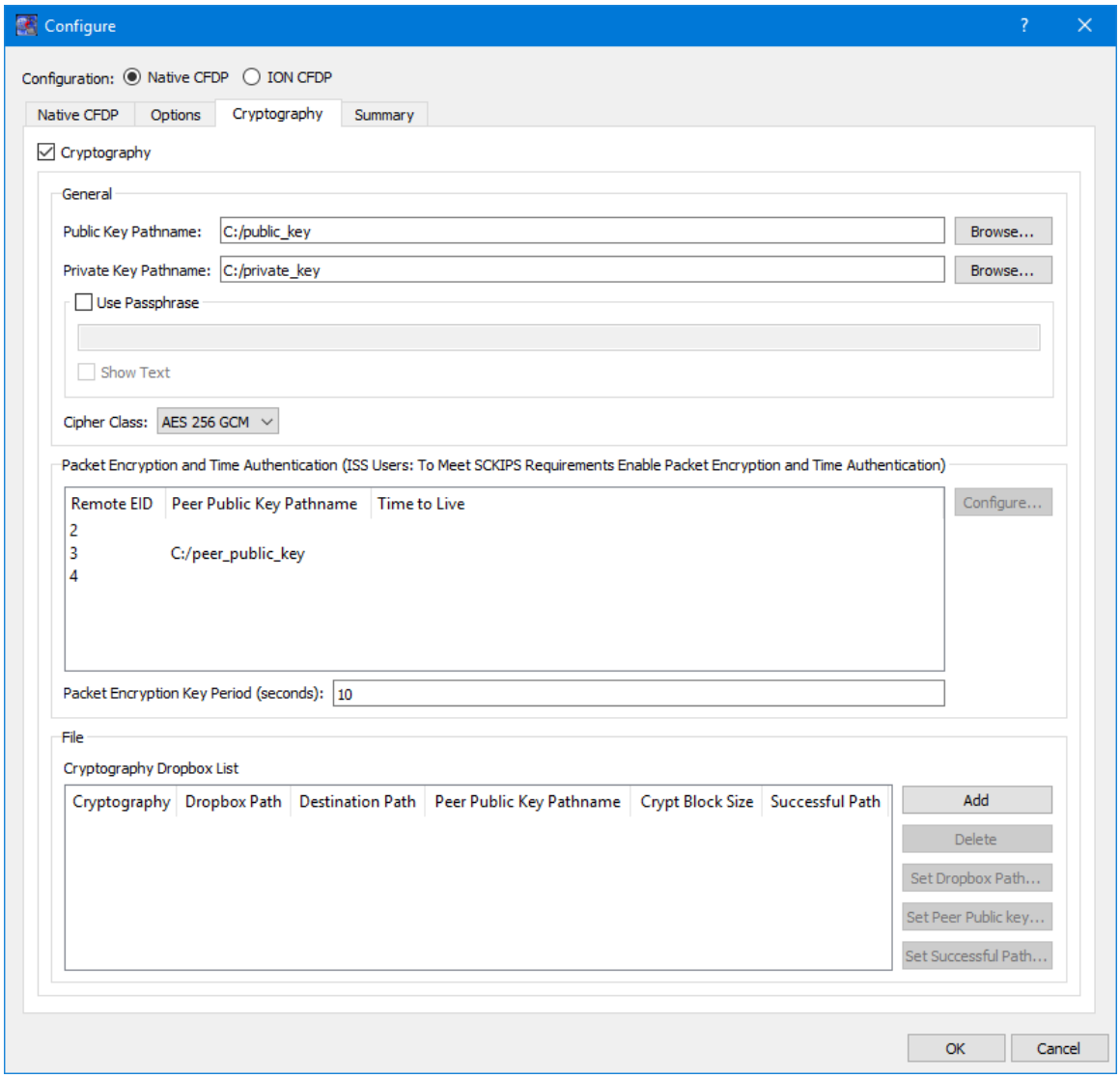

Figure 4 CFDP Cryptography Tab

The Packet list provides the capability to identify that cryptography services (encryption and decryption) should be used when communicating with a specific remote entity. All CFDP packets exchanged with the remote entity will be encrypted before being sent and decrypted upon reaching the destination when this feature is enabled. The Packet list will display all remote entities that were entered into the Remote Entities list on the General tab. If you do nothing all CFDP communication with the remote entities listed will be unencrypted. If you would like to use cryptography services when communicating with a remote entity you should provide a peer public key for the remote entity. You can select a remote entity in the list and then use the Configure button to assign a peer public key for that remote entity. Doing this will identify that you wish to use cryptography services when communicating with that remote entity. As you can see in Figure 4, peer public key information has only been entered for Remote EID 3. This means that communication with EID 2 and EID 4 will be unencrypted and communication with EID 3 will be encrypted and decrypted. As mentioned in the introduction, cryptography keys

are used to generate other keys behind the scenes. One of these keys is called a Cipher Encryption Key (CEK). The Packet Key Encryption Period defines how often to generate a new Cipher Encryption Key (CEK) when streaming encrypted data. It can be configured to generate a new CEK for a packet stream once every "x" seconds to support encryption of a high rate packet stream.

4.3 How to use TReK CFDP to Encrypt and Decrypt Files

The TReK CFDP application can be used to encrypt and decrypt files. This capability can be used stand-alone or combined with CFDP to encrypt a file before it is transmitted and then decrypt the file when it arrives at its destination. These capabilities can be used with Native CFDP and ION CFDP. Figure 5 shows the Cryptography Tab in the CFDP application. The Cryptography checkbox must be checked and the General section must be populated to use any cryptography features in the application.

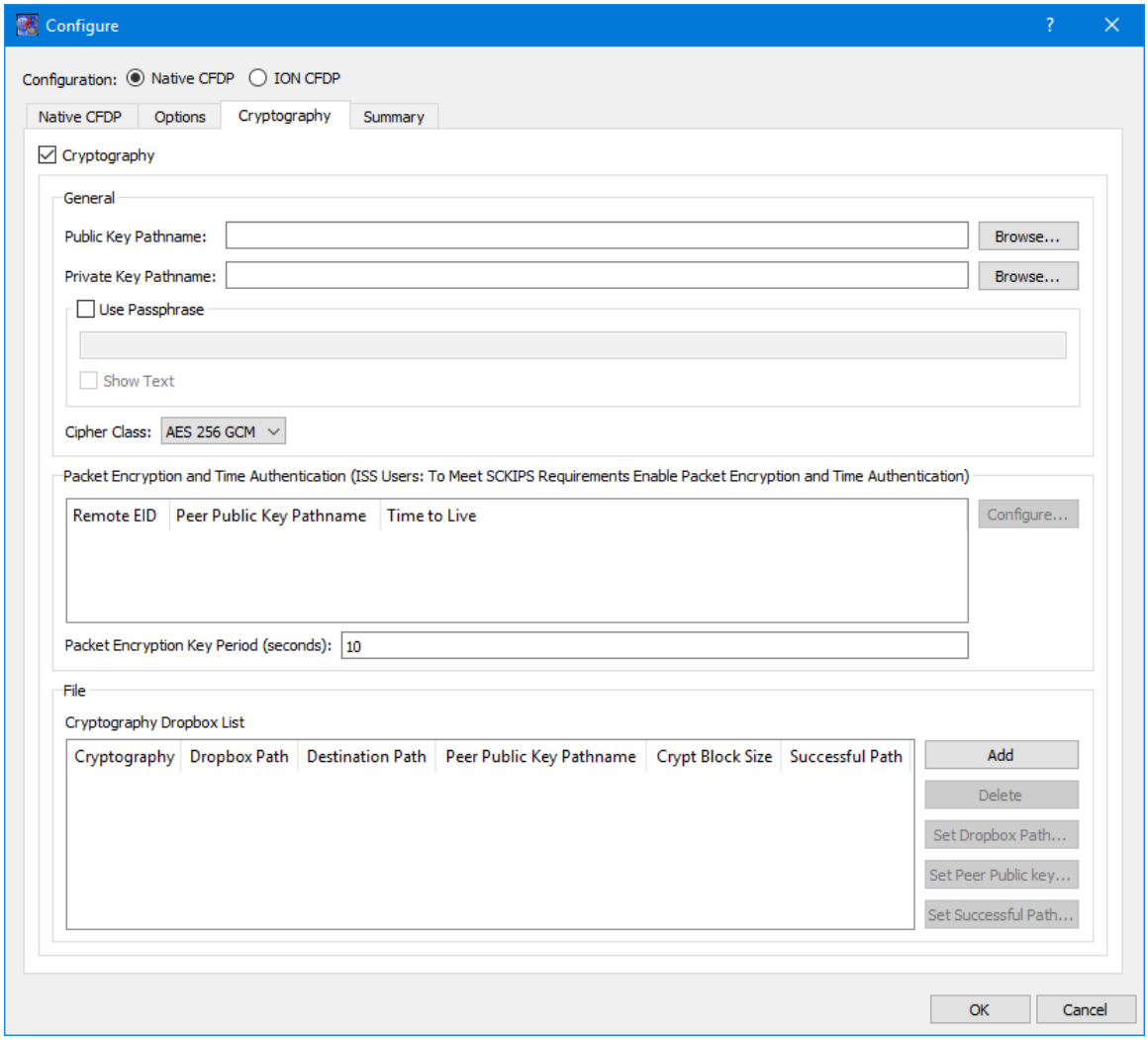

Figure 5 CFDP Cryptography Tab

General Cryptograpy Settings

The General section must be populated to use any cryptography features in the application. Each of the fields is described below:

Public Key Pathname

This is the absolute path for the public key file.

Private Key Pathname

This is the absolute path for the private key file.

Use Passphrase

The Use Passphrase checkbox is used to specify that the private key requires a passphrase to unwrap/decrypt the private key in the private key file. If this box is checked, the passphrase must be entered into the text field. The passphrase text will not be displayed in the clear. If you want to see the text entered in the clear, check the Show Text checkbox. This information is not stored when you save a configuration. You will have to enter it each time you restart the application when using cryptography services.

Show Text

The Show Text checkbox is used to display the passphrase text in the clear.

Cipher Class

This option menu is used to select the cipher class.

Packet Encryption and Time Authentication

The Packet Encryption and Time Authentication area provides the capability to identify that cryptography services (encryption and decryption) should be used when communicating with a specific remote entity. All CFDP packets will be encrypted before being sent and decrypted upon reaching the destination when this feature is enabled. This capability can only be used with Native CFDP. Please note that if you turn this feature on for a remote entity and you also define an encryption dropbox that will communicate with the remote entity, this will result in double encryption. The list will contain all remote entities that were entered into the Remote Entities list on the General tab. If you do nothing all CFDP communication with the remote entities listed will be unencrypted. If you would like to use cryptography services when communicating with the remote entity you should provide a peer public key for the remote entity. The public key from another computer is a peer public key. TReK will use peer public key terminology when referring to another computer's public key. To encrypt the packets for a specific remote entity you assign a peer public key for that remote entity. Doing this will identify that you wish to use cryptography services when communicating with that remote entity. In addition to encrypting the stream, you can also choose to use the time authentication feature to provide protection against inadvertent or malicious replay of the packets. The time authentication feature uses an encrypted timestamp, sequence count, and time to live value to provide replay resistance time authentication. The time authentication software compares the packet's decrypted time stamp with the operating system time (plus or minus the decrypted TTL) to determine if the packet's decrypted time falls within the authentication time window. In addition, the time authentication software does not allow

a packet to be processed if the decrypted time and sequence count stays the same or decreases when compared to the previous decrypted packet time and sequence count. These features can be configured for each Remote Entity by selecting the Remote Entity in the list and pushing the Configure button.

Packet Key Encryption Period

The Packet Key Encryption Period is only used when using the Packet encryption feature. As mentioned in the introduction, cryptography keys are used to generate other keys behind the scenes. One of these keys is called a Cipher Encryption Key (CEK). The Packet Key Encryption period defines how often to generate a new Cipher Encryption Key (CEK) when encrypting packet streams. It can be configured to generate a new CEK for a packet stream once every "x" seconds to support encryption of high rate playback stream. The time period is measured in seconds. If the packet key encryption period is set to zero, the TReK encryption library will generate a new packet encryption key for every packet in the stream. The TReK encryption library can support the encryption of high rate packet streams by setting the packet key encryption period to a non-zero value. The default value is 10 seconds.

File Cryptography Settings

The File section is used to define services to encrypt or decrypt files. It contains the Cryptography Dropbox List. The Cryptography Dropbox List is used to define one or more Cryptography dropboxes. There are two types of Cryptography dropboxes: encrypt and decrypt. An encrypt dropbox is used to encrypt a file. A decrypt dropbox is used to decrypt a file.

Dropbox configuration parameters include the type of cryptography dropbox, where the dropbox is located, the destination directory, the absolute path to the peer public key, the crypt block size, and the successful directory. Each parameter is described below:

Cryptography

Cryptography values are encrypt and decrypt. This identifies whether the dropbox is encrypting or decrypting files.

Dropbox Path

The Dropbox Path identifies the local directory to be used for the encrypt or decrypt dropbox.

Destination Path

The Destination Path identifies a local directory where the new encrypted or decrypted file is created and stored.

Peer Public Key Pathname

The Peer Public Key Pathname identifies the absolute path to the peer public key file. The peer public key is the public key of the destination platform.

Crypt Block Size

The Crypt Block Size is an unsigned 32 bit value identifying the number of bytes that are read and encrypted or decrypted with every file read. A large crypt block size improves encryption and decryption performance but may also tax a CPU.

Successful Path

If the Successful Path is defined, the dropbox will move the original file placed in the dropbox to the successful directory if and only if a new encrypted or decrypted file is successfully created and stored in the dropbox's destination directory. If the successful path is empty, the dropbox will delete the original file placed in the dropbox if and only if a new encrypted/decrypted file is successfully created and stored in the dropbox's destination directory. If the encrypt or decrypt dropbox fails to encrypt or decrypt a file, the file will be renamed with a time tagged ".droperror" extension and remain in the dropbox. The encrypt or decrypt dropbox will not attempt to encrypt or decrypt a file with a ".droperror" extension in its filename.

An encrypt or decrypt dropbox file is encrypted or decrypted prior to being transferred to a local destination directory on the dropbox platform. Pre-existing dropbox files are immediately encrypted or decrypted as soon as the CFDP service is activated. If the local destination directory of an encrypt dropbox is a CFDP dropbox, the encrypted file will automatically be transferred to the CFDP dropbox's remote destination directory. If the CFDP dropbox's remote destination directory is a decrypt dropbox the encrypted file will automatically be decrypted and placed in the decrypt dropbox's destination directory. By chaining together encrypt and decrypt dropboxes with a CFDP dropbox, a completely automated encrypt, CFDP file transfer, decrypt chain may be created and set in motion by placing a file in the local encrypt dropbox. The encrypt, decrypt, CFDP dropbox chain is currently the only method TReK provides to automate file encryption/decryption using ION CFDP.

Figure 6 shows an example of an encrypt dropbox definition. A file placed in the Dropbox Path (C:\encrypt_dropbox) will be encrypted using the peer public key identified and a 1,000,000 byte crypt block size. Since a Successful Path has been provided, the original file will be moved from the Dropbox Path directory to the Successful Path directory upon successful encryption.

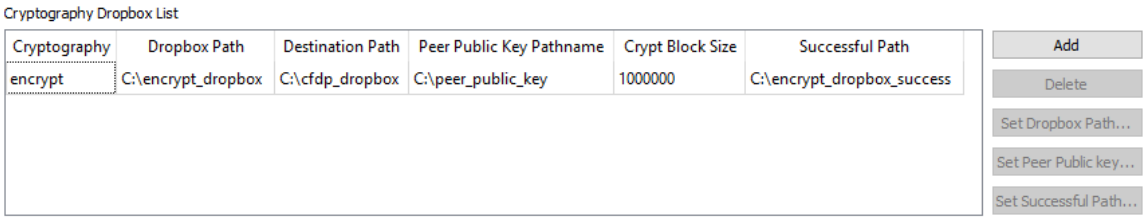

Figure 6 Encrypt Dropbox

Figure 7 and Figure 8 show how an encrypt dropbox and a CFDP dropbox can be used together to encrypt a file and then transfer it using CFDP. In Figure 7, the encrypt dropbox Destination Path points to the CFDP Dropbox Path. This “chains” the two dropboxes together. When a file is placed in the C:\encrypt_dropbox directory and

successfully encrypted, it will be placed in the C:\cfdp_dropbox folder and transferred to the CFDP dropbox Destination Path.

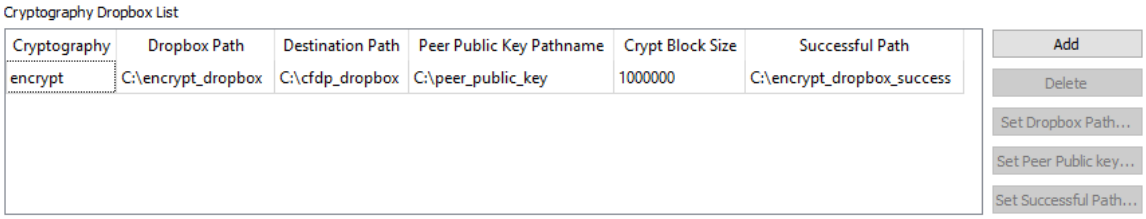

Figure 7 Encrypt Dropbox

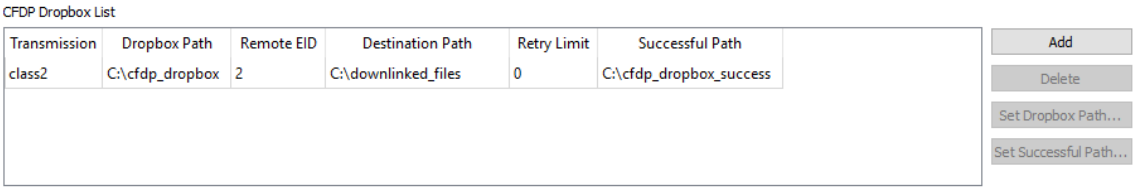

Figure 8 CFDP Dropbox

Once the encrypted file reaches its destination, you can leave it encrypted or decrypt it using a decrypt dropbox as shown in Figure 9.

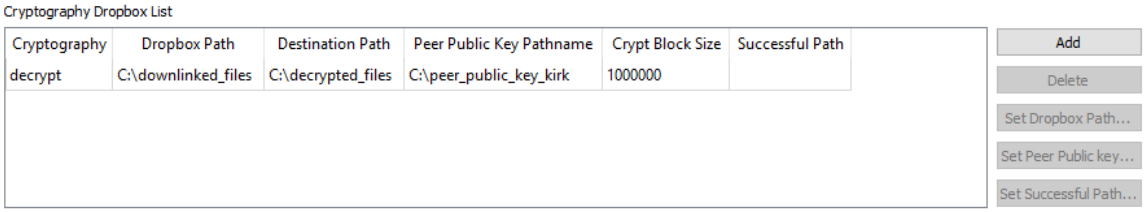

Figure 9 Decrpyt Dropbox

When the encrypted file arrives in the C:\downlinked_files directory on the remote computer decryption will be applied and if successful the decrypted file will be placed in the C:\decrypted files directory. There is no need to define a decrypt dropbox if you wish to leave the file encrypted when it reaches its destination. If a decrypt dropbox is not defined, the C:\downlinked_files directory would be the final destination for the encrypted file.

4.4 How to use TReK Data to receive encrypted data and decrypt it.

The TReK Data application provides the capability to receive encrypted data. You can choose to decrypt the incoming data and then process, record, or forward the decrypted data, or you can choose to leave the incoming data encrypted and record or forward the encrypted data. If you choose not to decrypt the data, processing is not available.

Configuring Cryptography Settings

To use cryptography features, you need to configure the application cryptography settings. These can be found in the Manage Cryptography Settings dialog available from the Options menu. The Manage Cryptography Settings dialog is shown in Figure 10. The Cryptography checkbox must be checked and all required cryptography settings must be populated with valid information in order to successfully activate a service that uses cryptography capabilities.

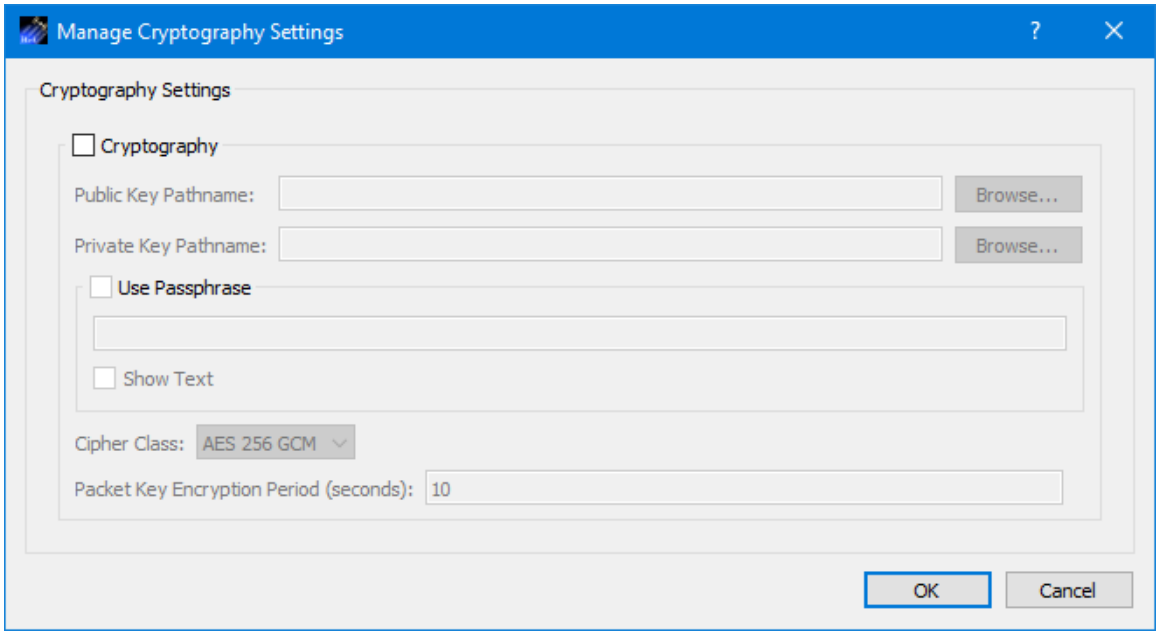

Figure 10 TReK Data Manage Cryptography Settings Dialog

Each of the fields is described below:

Cryptography Checkbox
The Cryptography checkbox is used to specify you want to use cryptography services.

Public Key Pathname
This is the absolute path for the public key file.

Private Key Pathname
This is the absolute path for the private key file.

Use Passphrase
The Use Passphrase checkbox is used to specify that the private key requires a passphrase to unwrap/decrypt the private key in the private key file. If this box is checked, the passphrase must be entered into the text field. The passphrase text will not be displayed in the clear. If you want to see the text entered in the clear, check the Show Text checkbox. This information is not stored when you save a configuration. You will have to enter it each time you restart the application when using cryptography services.

Show Text

The Show Text checkbox is used to display the passphrase text in the clear.

Cipher Class
This option menu is used to select the cipher class.

Packet Key Encryption Period
As mentioned in the introduction, cryptography keys are used to generate other keys behind the scenes. One of these keys is called a Cipher Encryption Key (CEK). The Packet Key Encryption period defines how often to generate a new Cipher Encryption Key (CEK) when streaming encrypted data. It can be configured to generate a new CEK for a packet stream once every "x" seconds to support encryption of high rate data stream.

Decrypting Incoming Data

Figure 11 shows the Add Service Data Source Cryptography Tab. This is how you configure the application to decrypt incoming data. This is done per service and per data source (data sender).

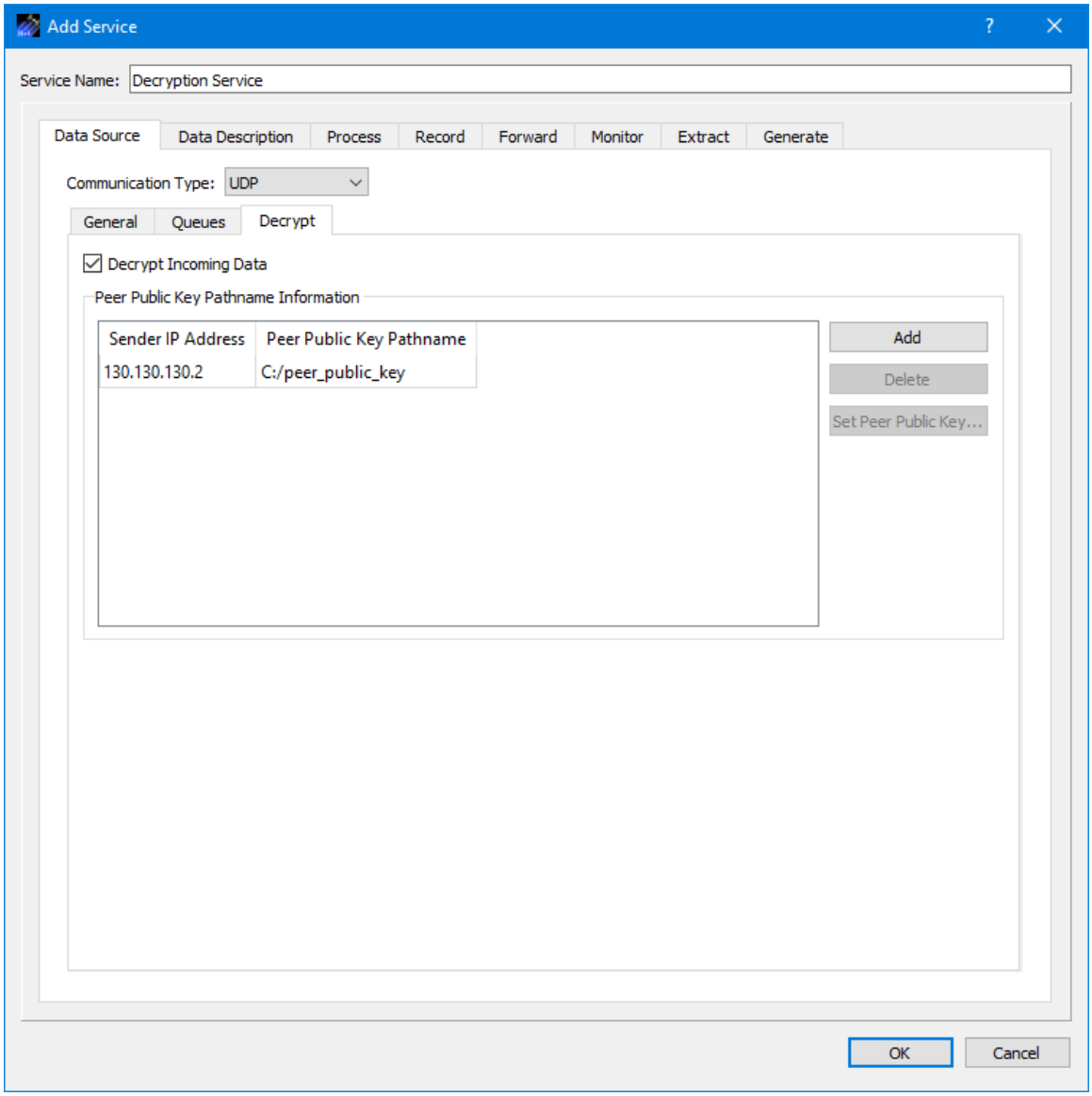

Figure 11 TReK Data Decryption Tab

To decrypt incoming data, the TReK software needs the IP address of the source of the data and the Peer Public Key for the encrypted data. It is possible to receive multiple encrypted data streams within one service. To decrypt incoming data, the Decrypt Incoming Data checkbox should be checked and the IP address and absolute path to the Peer Public Key associated with the data source should be entered for each source of incoming data. The Add button is used to add an entry. The Delete button is used to Delete an entry. The Peer Public Key button is used to browse the local disk for a peer public key file. One or more rows must be selected to use the Delete and Peer Public Key buttons.

If you are decrypting the data, the Data Description tab should be configured to identify the Packet Type of the decrypted data (e.g. PdssPayload).

Since the data will be decrypted, processing, recording, and forwarding options are available.

Working with Encrypted Data

If you choose to leave the incoming data encrypted, you should leave the Decrypt Incoming Data checkbox unchecked and configure the Data Description tab as shown in Figure 12.

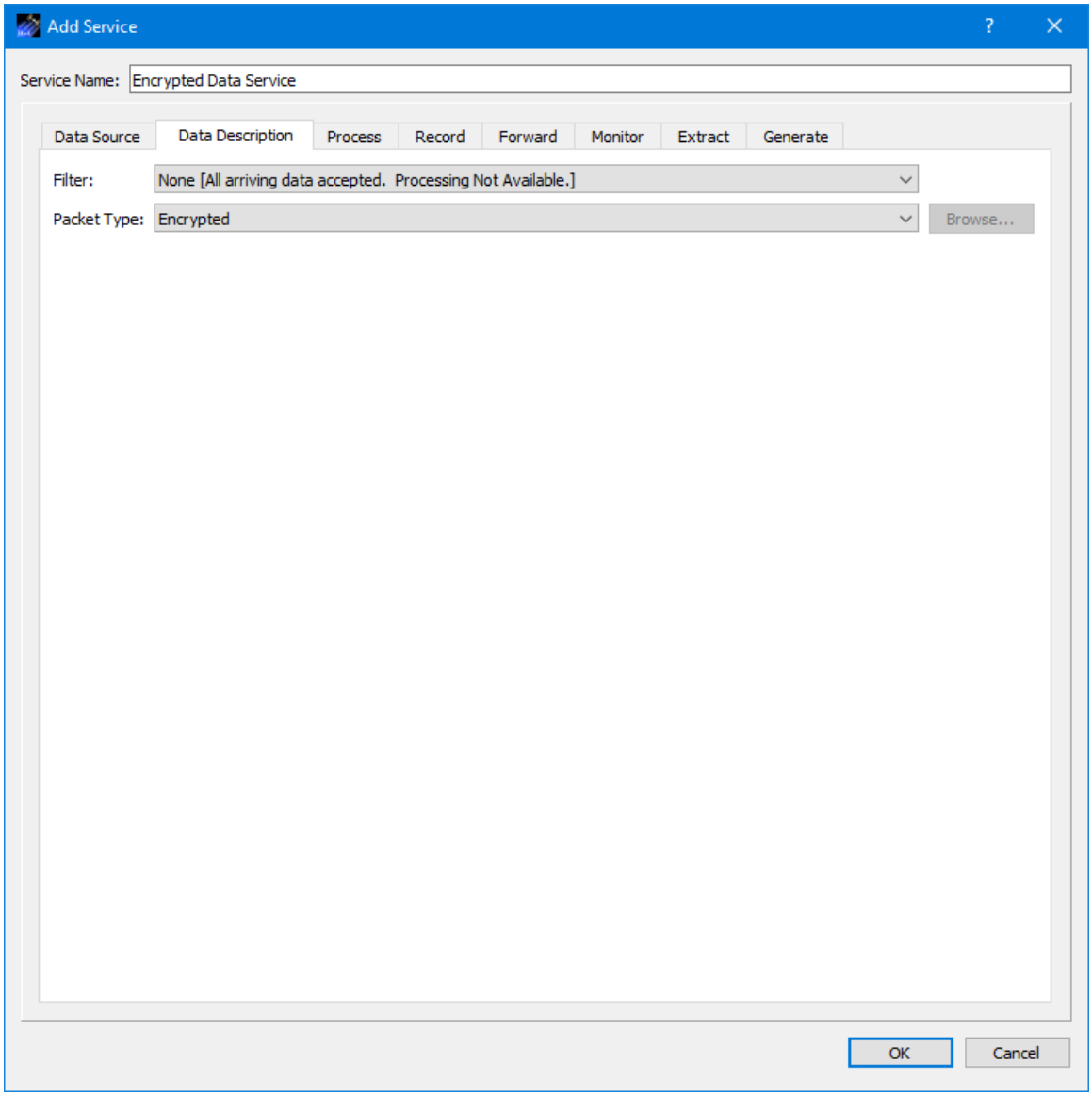

Figure 12 Data Description Tab

When the Filter Type is set to None, the Packet Type options are None and Encrypted. If you will be receiving encrypted data, and you do not want to decrypt the data, set the

Filter Type to None and the Packet Type to Encrypted. The incoming encrypted data can be recorded or forwarded.

4.5 How to use TReK Data to send encrypted data.

The TReK Data application provides the capability to encrypt data before forwarding it.

Configuring Cryptography Settings

To use cryptography features, you need to configure the application cryptography settings. These can be found in the Manage Cryptography Settings dialog available from the Options menu. The Manage Cryptography Settings dialog is shown in Figure 13. The Cryptography checkbox must be checked and all required cryptography settings must be populated with valid information in order to successfully activate a service that uses cryptography capabilities.

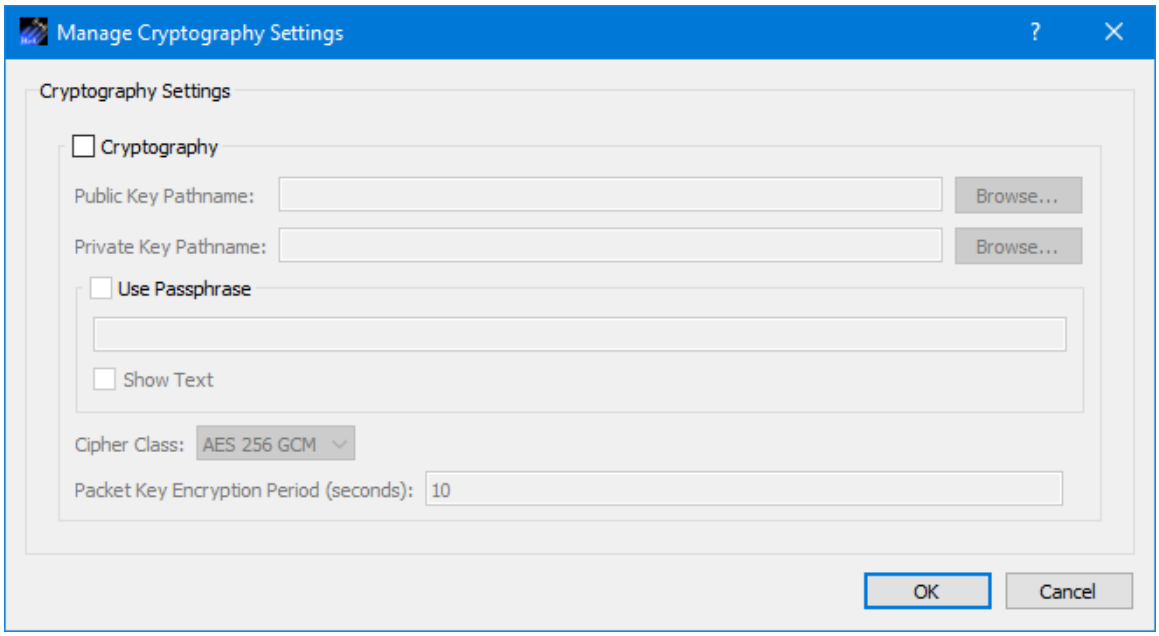

Figure 13 TReK Data Manage Cryptography Settings Dialog

Each of the fields is described below:

Cryptography Checkbox

The Cryptography checkbox is used to specify you want to use cryptography services.

Public Key Pathname

This is the absolute path for the public key file.

Private Key Pathname

This is the absolute path for the private key file.

Use Passphrase

The Use Passphrase checkbox is used to specify that the private key requires a passphrase. If this box is checked, the passphrase must be entered into the text field. The passphrase text will not be displayed in the clear. If you want to see the text entered in the clear, check the Show Text checkbox. This information is not stored when you save a configuration. You will have to enter it each time you restart the application when using cryptography services.

Show Text
The Show Text checkbox is used to display the passphrase text in the clear.

Cipher Class
This option menu is used to select the cipher class.

Packet Key Encryption Period
As mentioned in the introduction, cryptography keys are used to generate other keys behind the scenes. One of these keys is called a Cipher Encryption Key (CEK). The Packet Key Encryption period defines how often to generate a new Cipher Encryption Key (CEK) when encrypting packet streams. It can be configured to generate a new CEK for a packet stream once every "x" seconds to support encryption of high rate data stream.

Forwarding Encrypted Data

Figure 14 shows the Add Service Forward Tab Destination List and Figure 15 shows the Forward Tab Encrypt Tab. The Encrypt Tab is used to identify which destinations should receive encrypted data. Encryption is controlled per service and per destination.

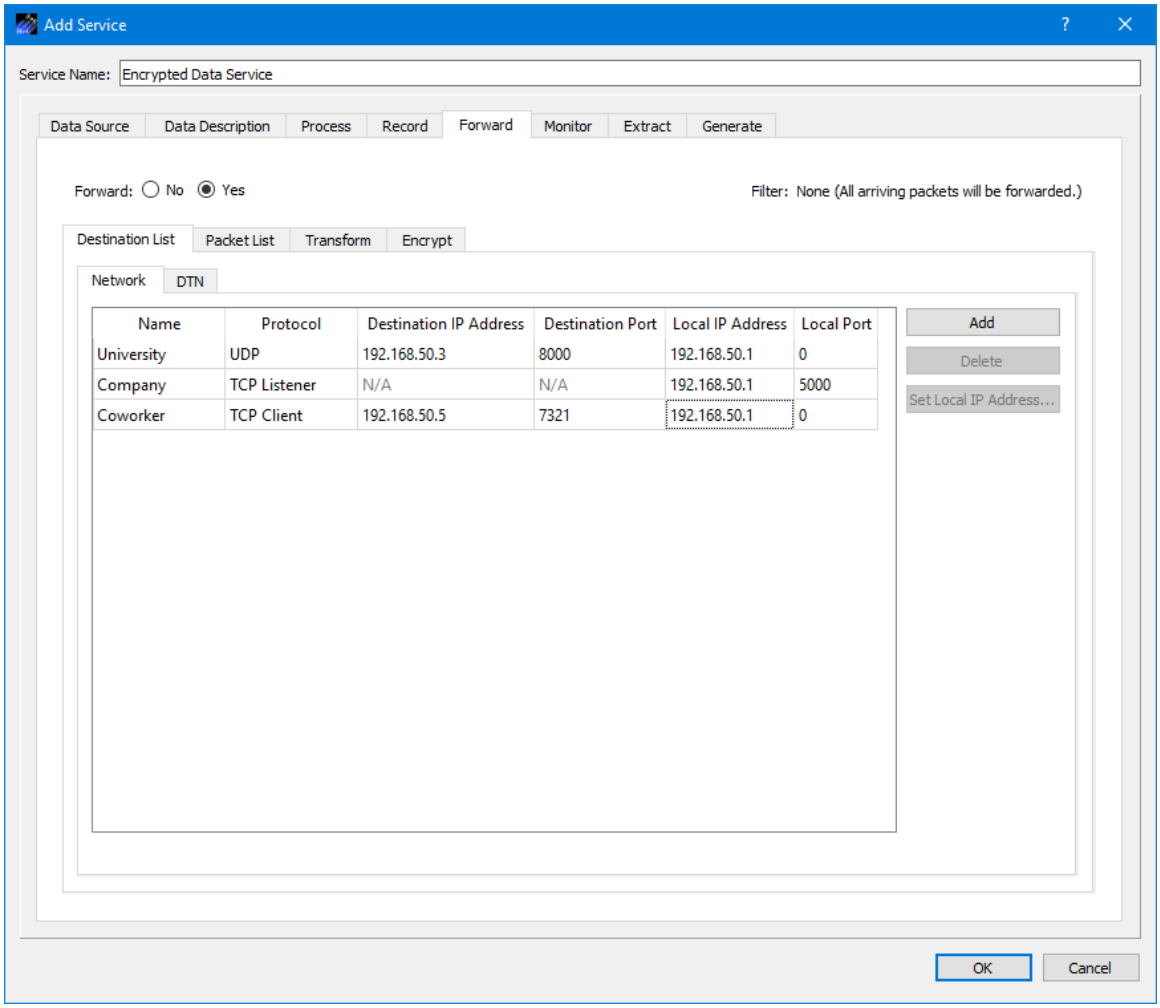

Figure 14 TReK Data Forward Tab Destination List

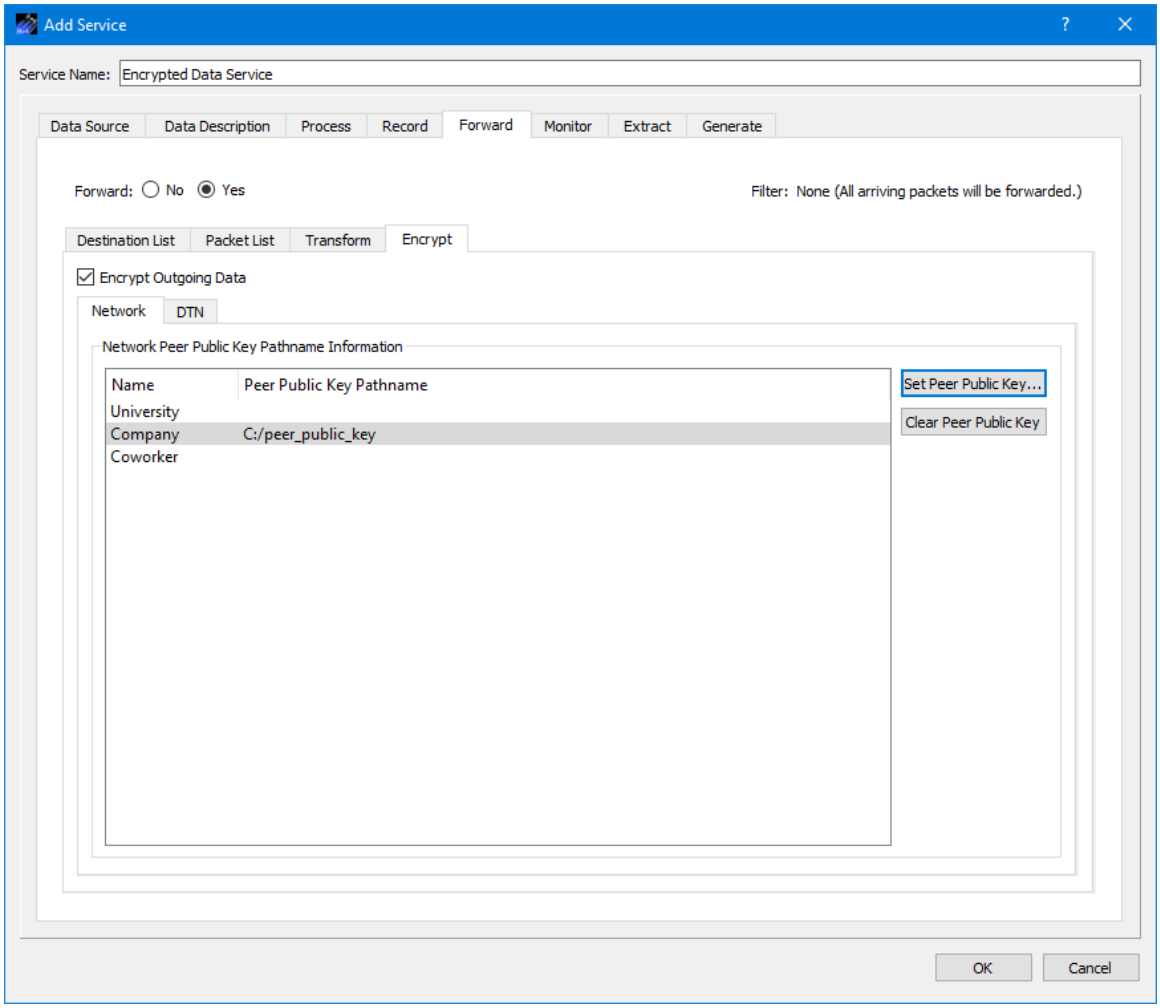

Figure 15 TReK Data Forward Tab Encrypt Tab

The Peer Public Key Pathname Information list will display the destinations that were entered on the Destination List tab. To encrypt data going to a specific destination, check the Encrypt Outgoing Data checkbox and identify the absolute path to the Peer Public Key associated with that destination. If no Peer Public Key Pathname is identified, the data forwarded to that destination will not be encrypted (even if the Encrypt Outgoing Data checkbox is checked). The Set Peer Public Key button is used to browse the local disk for a peer public key file. The Clear Peer Public Key button is used to clear a peer public key pathname. One or more rows must be selected to use the Set Peer Public Key and Clear Peer Public Key buttons.

In Figure 15, only data going to the Company destination will be encrypted.

4.6 How to use TReK Data to record encrypted data.

The TReK Data application can be used to record encrypted data that it receives. When encrypted data is recorded, the configuration file describing the recorded data file's

format will identify that the data is encrypted. The TReK Playback application can be used to play back data in an encrypted recorded data file.

If you have unencrypted data that you would like to encrypt and then record, you can configure two services in the TReK Data application to perform this task. Configure one service to receive the unencrypted data and forward it as encrypted data. Then configure a second service to receive and record the incoming encrypted data.

4.7 How to use TReK Playback to play back Encrypted Recorded Data Files

The TReK Playback application can be used to play back data in an encrypted recorded data file. This can be configured using the Configure dialog in the TReK Playback application. Figure 16 shows the Configure dialog with an encrypted recorded data file in the list.

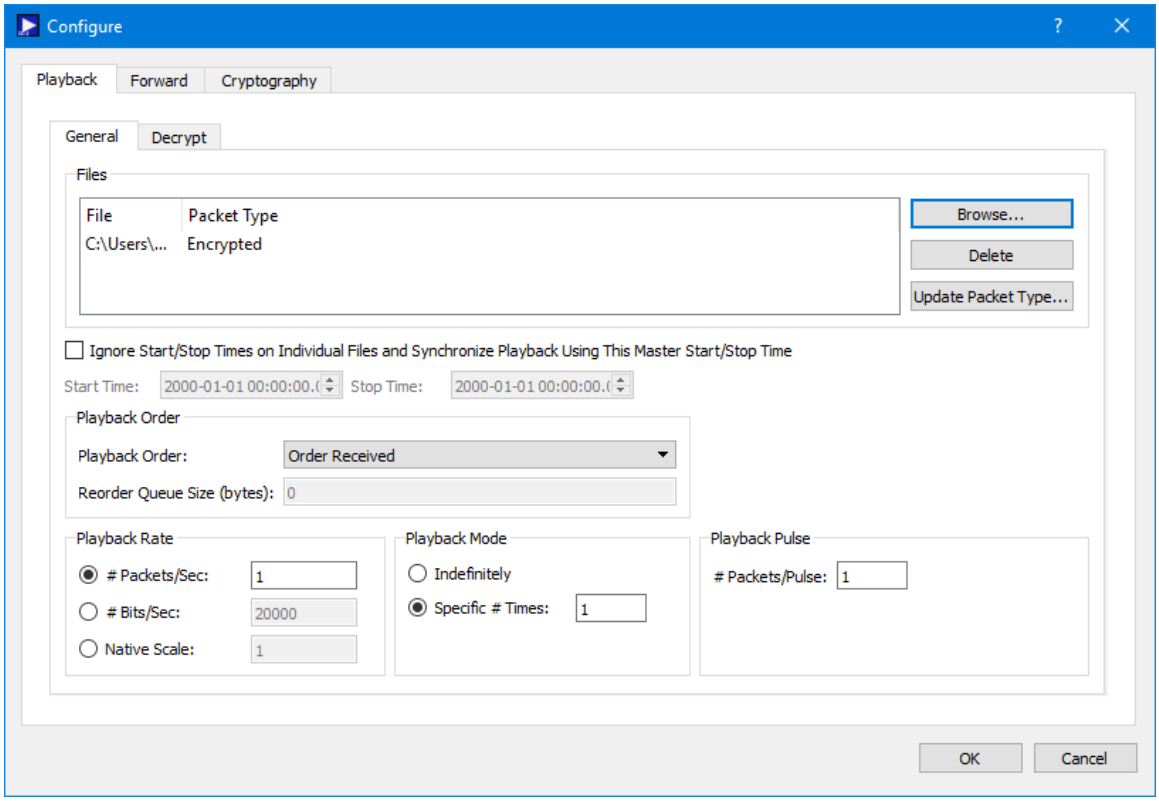

Figure 16 TReK Playback Configure Dialog

Note that the Packet Type shown is Encrypted. This information was derived from the recorded data configuration file. If you select a recorded data file that does not have an associated configuration file, you will need to use the Update Packet Type dialog to set the Packet Type to the correct setting. The following scenarios describe what default packet type will be displayed based on several different factors and whether you need to take any action to update the packet type.

Scenario 1

Recorded Data File Contains:	PdssPayload 7 Data
Is Data Encrypted?	No
Does Recorded Data Configuration File Exist?	Yes
When File is Added Packet Type Displayed:	PdssPayload

Action:
No Action is necessary since the default packet type displayed is correct. The data being played back is unencrypted PdssPayload data.

Scenario 2

Recorded Data File Contains:	PdssPayload 7 Data
Is Data Encrypted?	Yes
Does Recorded Data Configuration File Exist?	Yes
When File is Added Packet Type Displayed:	Encrypted

Action:
No Action is necessary since the default packet type displayed is correct. The data being played back is encrypted data.

Scenario 3

Recorded Data File Contains:	PdssPayload 7 Data
Is Data Encrypted?	No
Does Recorded Data Configuration File Exist?	No
When File is Added Packet Type Displayed:	UserDefined

Action:
Use the Update Packet Type dialog to change the packet type to identify the type of data in the recorded data file (PdssPayload) since the data being played back will be unencrypted PdssPayload data.

Scenario 4

Recorded Data File Contains:	PdssPayload 7 Data
Is Data Encrypted?	Yes
Does Recorded Data Configuration File Exist?	No
When File is Added Packet Type Displayed:	UserDefined

Action:
Use the Update Packet Type dialog to change the packet type to identify the type of data in the recorded data file (Encrypted) since the data being played back will be encrypted data.

Decrypting Data

If you want to decrypt the encrypted data that is read from the file before forwarding the data, you can use the Decrypt Tab shown in Figure 17.

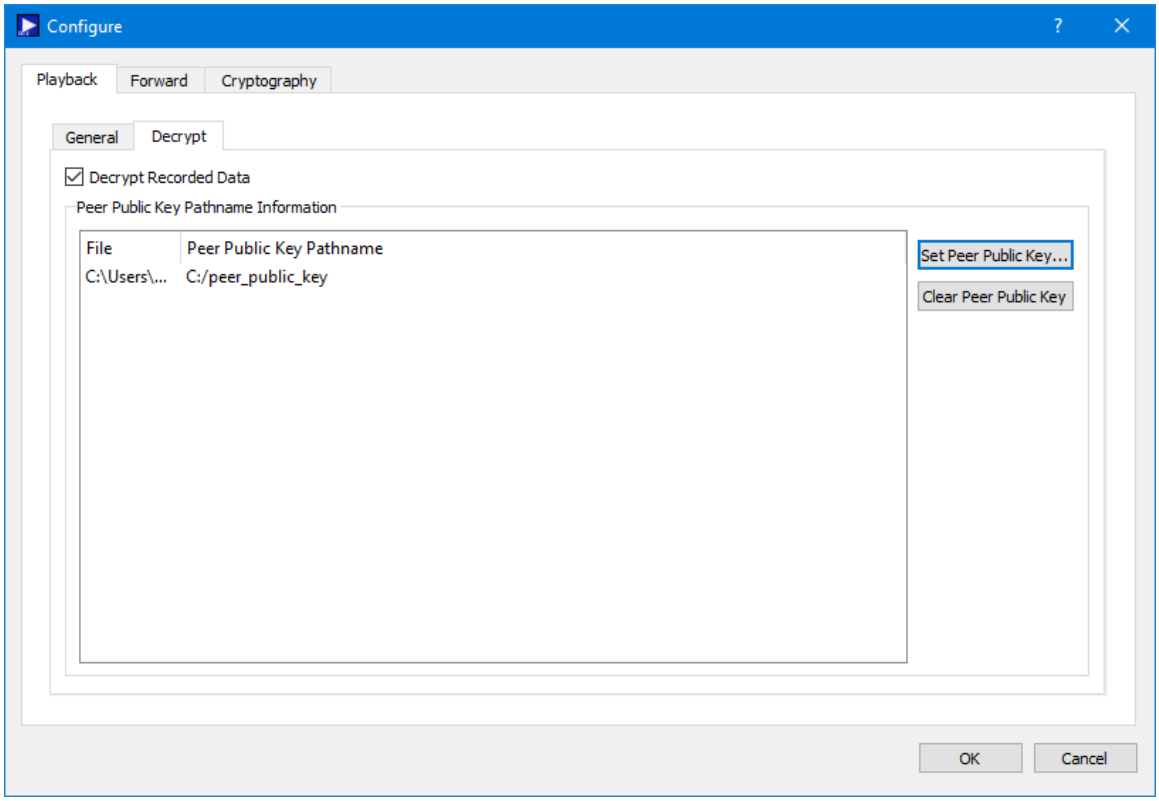

Figure 17 TReK Playback Configure Dialog Decrypt Tab

To decrypt the data, check the Decrypt Recorded Data checkbox and identify the peer public key for each recorded data file that you want to decrypt. Data read from the file will be decrypted before it is forwarded.

Forwarding Encrypted Data

If you want to encrypt the data before it is forwarded, you can use the Encrypt tab on the Forward Tab as shown in Figure 18. It is possible to "double" encrypt data by playing back encrypted data and then encrypting the data before it is forwarded. It is also possible to play back encrypted data, decrypt the data, and then encrypt the data before it is forwarded. It is doubtful either of these scenarios are useful but they are possible.

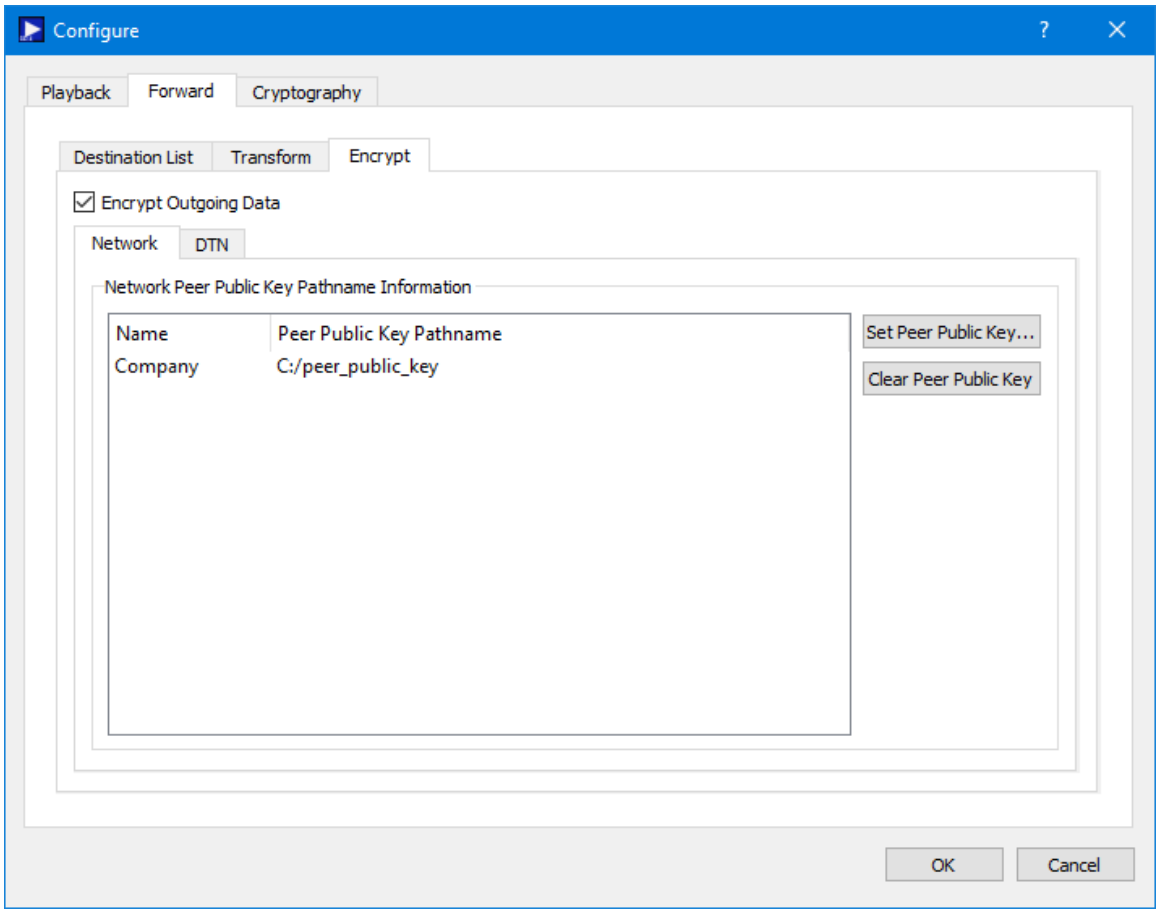

Figure 18 TReK Playback Forward Encrypt Tab

Configuring Cryptography Settings

If you are working with encrypted data or use any of the cryptography services (decrypt or encrypt) in the Configure dialog you need to fill out the Cryptography tab shown in Figure 19. The Cryptography checkbox must be checked and all required cryptography settings must be populated with valid information in order to successfully activate a playback that uses cryptography capabilities.

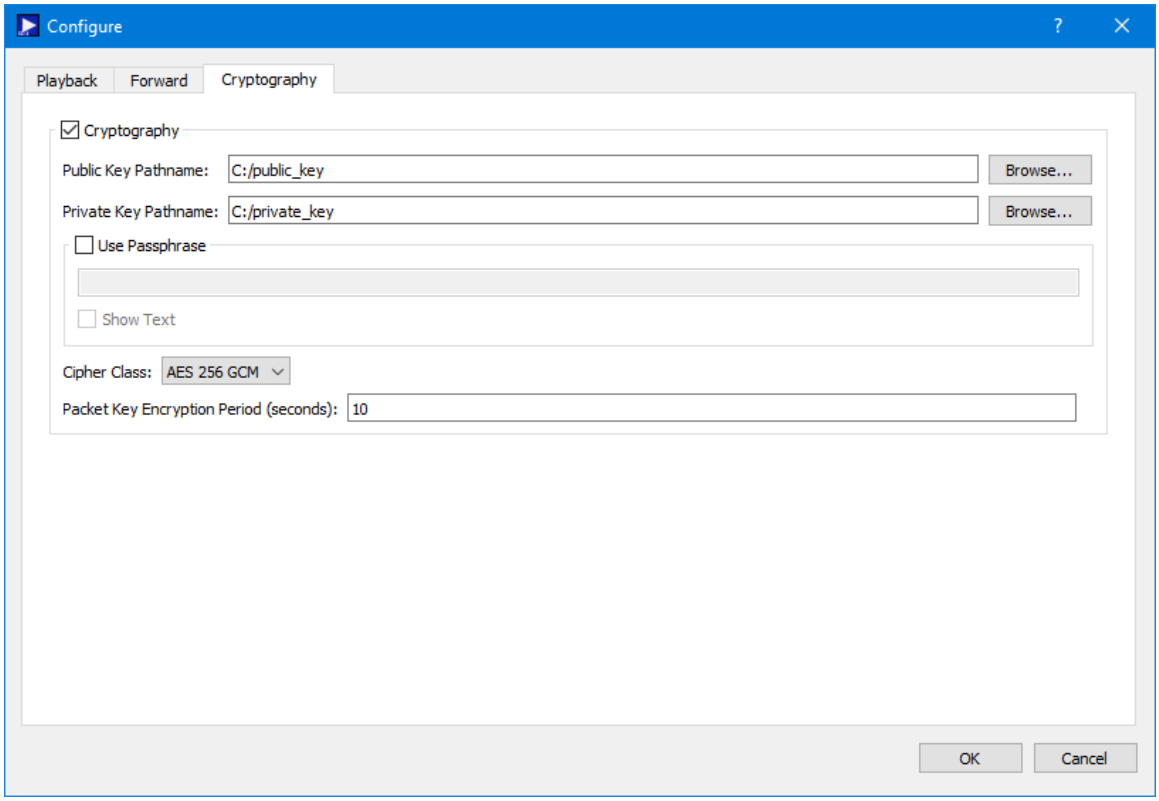

Figure 19 TReK Playback Configure Dialog Cryptography Tab

Each of the fields is described below:

Cryptography Checkbox
The Cryptography checkbox is used to specify you want to use cryptography services.

Public Key Pathname
This is the absolute path for the public key file.

Private Key Pathname
This is the absolute path for the private key file.

Use Passphrase
The Use Passphrase checkbox is used to specify that the private key requires a passphrase to unwrap/decrypt the private key in the private key file. If this box is checked, the passphrase must be entered into the text field. The passphrase text will not be displayed in the clear. If you want to see the text entered in the clear, check the Show Text checkbox. This information is not stored when you save a configuration. You will have to enter it each time you restart the application when using cryptography services.

Show Text
The Show Text checkbox is used to display the passphrase text in the clear.

Cipher Class

This option menu is used to select the cipher class.

Packet Key Encryption Period

As mentioned in the introduction, cryptography keys are used to generate other keys behind the scenes. One of these keys is called a Cipher Encryption Key (CEK). The Packet Key Encryption period defines how often to generate a new Cipher Encryption Key (CEK) when playing back encrypted data. It can be configured to generate a new CEK for a packet stream once every "x" seconds to support encryption of high rate playback stream. The time period is measured in seconds. If the packet key encryption period is set to zero, the TReK encryption library will generate a new packet encryption key for every packet in the stream. The TReK encryption library can support the encryption of high rate packet streams by setting the packet key encryption period to a non-zero value. The default value is 10 seconds.

4.8 How to use TReK CFDP Console to Encrypt and Decrypt all CFDP Communication

The TReK CFDP Console application can be used to encrypt and decrypt all CFDP communication, making it possible to configure the CFDP Console application to encrypt and decrypt all CFDP transactions (e.g., "put", "get", "message", "create_file", "delete_file" ...). All packets are encrypted prior to transmission and decrypted upon receipt. This capability can only be used with Native CFDP and is implemented in the Native CFDP configuration section of the TReK CFDP Console application's configuration file. Review the following CFDP configuration file parameters in the TReK CFDP Console User Guide or TReK Online Help for further information on this Native CFDP encrypt/decrypt configuration option.

public_key_path_and_file_name	The public key path and filename is the absolute path and file name of the local entity's public key file. It is used to encrypt and decrypt files and CFDP PDU packets. The public key file is created by TReK's "trek_crypt" program.
private_key_path_and_file_name	The private key path and filename is the absolute path and file name of the local entity's private key file. It is used to encrypt and decrypt files and CFDP PDUs packets. The private key file is created by TReK's "trek_crypt" program.
packet_encryption_key_time_interval	The packet encryption key time interval determines how often the packet encryption key is changed while encrypting a stream of native CFDP PDU packets. The time interval is measured in seconds. If the packet encryption key time interval is set to zero, the TReK encryption library will generate a new packet encryption key for every packet in the stream. The

	TReK encryption library can support the encryption of high rate packet streams by setting the packet encryption key time interval to a non-zero value. The default value is 10 seconds.
cipher_class	The cipher class is the cipher package that the TReK encryption library will use to encrypt and decrypt files and streams of native CFDP PDU packets. The four cipher class values are AES_128_GCM, AES_256_GCM, AES_128_CCM and AES_256_CCM which support either a 128 bit or 256 bit symmetric key. An AES 256 cipher will require more CPU resources to encrypt and decrypt files and streams then an AES 128 cipher.
remote_entity_id remote_ip_address remote_port peer_pub_key_path_and_file_name	The pre-assigned remote entity ID integer value and its associated remote IP address, remote port and the absolute path and name of the file containing the peer public key to encrypt and decrypt the native CFDP PDU packets. The peer public key is the public key of the remote/destination platform and is created by TReK's "trek_crypt" program. A peer public key path and file name must be provided to enable encryption and decryption of all CFDP transactions with the remote entity. Multiple remote entity ID entries are supported by the CFDP library.
remote_entity_id remote_ip_address remote_port peer_pub_key_path_and_file_name time_to_live	The pre-assigned remote entity ID integer value and its associated remote IP address, remote port and the absolute path and name of the file containing the peer public key to encrypt and decrypt the native CFDP PDU packets. The peer public key is the public key of the remote/destination platform and is created by TReK's "trek_crypt" program. A peer public key path and file name must be provided to enable encryption and decryption of all CFDP transactions with the remote entity. If the encrypted CFDP session requires an encrypted timestamp, sequence count and TTL value to provide replay resistance time authentication in support of SCKIPS, include the TTL value, in seconds, after the peer public key path and file name. Multiple remote entity ID entries are supported by the CFDP library. Minimum TTL value is 0, maximum TTL value is 65535 and the default TTL value is 60.

Table 1 TReK CFDP Configuration File Parameters

4.9 How to use TReK CFDP Console to Encrypt and Decrypt Files

The TReK CFDP Console application can be used to encrypt and decrypt files. This capability can be used stand-alone or combined with CFDP to encrypt a file before it is transmitted and then decrypt the file when it arrives at its destination. These capabilities can be used with Native CFDP and ION CFDP. The TReK CFDP Console application performs file encryption and decryption by creating cryptography dropboxes. An encrypt dropbox is used to encrypt a file. A decrypt dropbox is used to decrypt a file.

A dropbox primitive in the TReK CFDP Console application's configuration file defines the parameters of the dropbox and includes the type of cryptography dropbox, where the dropbox is located, the destination directory, the absolute path to the peer public key, the crypt block size, and the successful directory. Acceptable formats of the encrypt or decrypt dropbox primitive string are as follows:

- dropbox <encrypt/decrypt> <"dropbox path"> <" peer public key path and filename"> <"destination path"> <crypt block size> <"successful transaction path"> (e.g., dropbox encrypt "D:/dropbox_dest1/" "D:/ peer_public.key" "D:/dest1/" 10000 "D:/success/")

- dropbox <encrypt/decrypt> <"dropbox path"> <" peer public key path and filename"> <"destination path"> <crypt block size> <""> (e.g., dropbox decrypt "/home/user/dropbox_dest1" "/home/user/ peer_public.key" "/home/user/dest1" 10000 "")

Review the "How to Configure the Application" and "How to Create an Encrypt or Decrypt Dropbox" descriptions in the TReK CFDP Console User Guide for further information on this configuration file option.

4.10 How to use TReK Device Services API to Encrypt and Decrypt Packets or Bundles

The TReK Device Services API may be included in a user developed application to encrypt and decrypt packets or ION bundles that flow across TReK sockets or TReK ION devices created by the user application. The TReK Device Services API provides ANSI C functions to generate public and private key pairs using the GeneratePublicAndPrivateKeyPair or GeneratePublicAndPrivateKeyPairWithPassphrase functions. The API also provides the AddCipherToDevice function to enable TReK socket encryption and decryption of packets or TReK ION device encryption and decryption of bundles. The API's SendPacket and ReceivePacket functions allow a user application to communicate with the encryption and decryption sockets and ION devices. Review the TReK's Online Help description of the Device Services API for further information on these API functions.