



IBM iSeries Tape Security

With Alliance PGP Encryption

Quick Start Guide

Version 1.01
March 15, 2005

Copyright Patrick Townsend & Associates, Inc 2005
All rights reserved

Introduction

This Quick Start guide discusses how to secure the data you store on tape using the Alliance PGP encryption product. While the focus of this paper is on tape security, the concepts presented here would apply to most off-line storage media.

There are two basic reasons why most companies copy iSeries data to tape:

To share the data with a trading partner such as a bank, benefits provider, payroll processing company, or marketing service provider. Or,

To back up your iSeries data for archival and use in a disaster recovery event, or to back up your iSeries data for regulatory compliance.

Each of these have different requirements for saving and restoring the data in a secure manner. When sending a tape to a trading partner, you need to be sure the trading partner can process the tape on their computer system which is often not an iSeries system. And you will usually only be sending an extract of data that the trading partner needs.

However, when you back up your iSeries for archival or regulatory compliance needs you are usually saving an entire library or set of files. If the data needs to be restored it will be restored directly to your iSeries system, or to a business recovery system on your site or at a BRS vendor site. You will always be using native iSeries commands to save and restore the data.

In the sections below each of these scenarios are discussed with examples of the encryption and decryption operations.

For information on the Alliance encryption products please contact Patrick Townsend & Associates, Inc. or your Alliance software partner. You can contact Patrick Townsend & Associates at:

Web:	http://www.patowndsend.com
Phone:	(360) 357-8971 (800) 357-1019
International:	+1 360 357 8971
Fax:	(360) 357-9047
Email:	Info@patowndsend.com

Examples in this QuickStart Guide

The following examples are included in this QuickStart guide.

Example 1 - Encrypting a database file for tape transfer to a trading partner

Example 2 – Encrypting an IFS file for tape transfer to a trading partner

Example 3 – Encrypting saved objects for backup archival

Example 4 – Encrypting a library for tape archival

Example 5 – Restoring and decrypting a file from a trading partner

Example 6 – Restoring and decrypting a library from tape archival

This list of examples is not intended to be exhaustive, and there are other uses for Alliance PGP encryption to tape. However, the examples try to show typical uses of Alliance PGP encryption when used with tape. Please consult the Alliance PGP encryption reference guide for more information on how to use the Alliance encryption and decryption commands.

Tape concepts

The iSeries platform supports many different types of tape drives and media. Many iSeries systems have a built-in tape drive that supports QIC formatted tapes. Others have stand-alone tape systems that use the 3480 or 3490 formatted tapes cartridges. And there are many other types of tape drives and tape formats. Many of these tape drives and formats are supported on non-iSeries platforms such as UNIX (AIX, HP-UX, Solaris, etc.), Linux, Windows, and other platforms

When saving a file for a trading partner you will normally want to use standard label tapes and copy the data using the Copy To Tape (CPYTOTAP) command. The CPYTOTAP command creates a standard label tape that many other types of computer systems can read. The CPYTOTAP command has many options on formatting and blocking the data on the tape. Your trading partner can tell you which options to use when creating a tape.

When saving a file or library for backup or archival, you will normally use standard label tapes and copy the data to tape using the Save Library (SAVLIB) or Save Object (SAVOBJ) command. The data on the tape is formatted in Save-Restore format which can only be processed on an iSeries computer system.

In both cases you will want to secure the data with strong encryption to prevent unauthorized access through loss or theft. Alliance PGP Encryption will provide that data security.

Securing tape files for a trading partner

When you send a tape to a trading partner you can secure the data with Alliance PGP Encryption before copying the data to tape. You will prepare the file to send to the trading partner in an iSeries library, encrypt the file with Alliance PGP encryption, then save the encrypted file from a library to tape.

PGP encrypted files can be in one of two formats: Binary encrypted format, or ASCII armored format. For the purposes of saving a file to tape you will use the ASCII Armor format. This format is easy to store in an iSeries library, transfers easily to tape using a fixed record length and fixed block length format, and is easy to restore by the recipient.

When you send an encrypted file to a trading partner they will need to use a PGP product to decrypt the file. That PGP product can run on any computer system and does not need to be Alliance PGP Encryption.

The following steps would be involved in sending a tape to a trading partner:

- You create a PGP public and private key on your iSeries platform using Alliance PGP Encryption. This is a one time installation task.
- Your trading partner exports their PGP public key and sends it to you. This is done one time.
- You add the trading partner's key to your Alliance PGP key ring. This is done one time for each trading partner.
- You sign your trading partner's key on the Alliance PGP key ring. This indicates that you trust the partner's key. This is done one time for each trading partner.
- If you wish to sign the encrypted file with your private PGP key, you export your public PGP key and send it to your trading partner. Signing a file insures data integrity and authenticates you as the sender. It is an optional step with PGP encryption. This is done one time for each trading partner.
- You prepare the data that you want to send to your trading partner. You may need to unpack numeric fields or format the file in CSV or tab-delimited format. You can use the Alliance CrossData/400 product to do this. The prepared data resides in an iSeries library in the EBCDIC character set. This is done each time you want to send a file to a trading partner.
- You encrypt the data with Alliance PGP Encryption specifying the ASCII Armor option, and the option to copy the encrypted data to a new iSeries file. You can specify that the data be converted to the ASCII character set prior to encryption, or remain in the EBCDIC character set. The result is a compressed, encrypted file in an iSeries library. The file may be in the ASCII character set, or in the EBCDIC character set.
- You then save the file to tape using the CPYTOTAP command, and send it to your trading partner.

This is the general process used for securing data to send to a trading partner. The examples below show more detail on how the process works.

Securing tape files for backup and archival

The process for securing files on a backup archival tape is a bit different. You are normally saving an entire library or set of objects to tape, and these objects are in a save file or Save-Restore format. The tapes will always be processed on an iSeries system, so there are no issues related to cross-platform processing of the tapes.

A save for archival purposes contains many different types of objects, and not just files. A typical save would include programs, data areas, database files, data queues, and perhaps many other types of objects. And the objects do not need to be converted to another format prior to the save. They are saved directly from the iSeries library.

Alliance PGP Encryption supports securing backup tapes by supporting the encryption of save files in an iSeries library. The save file is encrypted to an IFS directory, and the copied back to a library in compressed ASCII Armor format to be saved to tape.

The following steps would be involved in creating a tape for archival:

- You create a PGP public and private key on your iSeries platform using Alliance PGP Encryption. This is a one time installation task.
- You save the library or objects to a save file on the iSeries platform. You can use the Save Objects (SAVOBJ) command, the Save Library (SAVLIB) command, Save Changed Objects (SAVCHGOBJ) command, or the Save (SAV) command. The result is one or more objects or libraries in a save file on the iSeries.
- You encrypt the save file with Alliance PGP Encryption specifying the ASCII Armor option, and the option to copy the encrypted data to a new iSeries file. You will specify that no data conversion be performed during the encryption process in order to protect the integrity of the save file data. The result is a compressed, encrypted file in an iSeries library in the compress ASCII Armor format.
- You then save the file to tape using the Save Object (SAVOBJ) command and send the tape to your archival vendor.

Requesting a secure tape from a trading partner

When you request a tape from a trading partner, and they do not have an iSeries system, you will probably want to ask for the following options:

- PGP encrypted with ASCII armor
- Copy to tape using standard labels, fixed record length, and fixed block length. PGP ASCII armored files are normally 64 or 65 characters in length. If each record in the file has a single line feed character, the record length will be 64. If each record in the file has a line feed and carriage return characters, the record length will be 65
- In order to encrypt the file for you, you will need to send the trading partner your PGP public key. Use the Alliance PGP option to export the key to an IFS directory. You can then copy the key to your PC and send it to your partner.

When you receive the tape you can use the Copy From Tape File (CPYFRMTAPF) command to restore the tape. The CPYFRMTAPF command has several options that let you specify record length and other options.

Since tape formats vary from one system to another you should expect to spend some time exchanging tapes and testing the file transfer from tape to your iSeries database.

Using PGP additional decryption keys (ADK)

Alliance PGP supports the use of Additional Decryption Keys. This means that you can encrypt a file for multiple recipients. This feature is not supported by products that are built on OpenPGP source code distributions. When using Alliance PGP to encrypt files for tape this may be a feature that can help you work with your Business Recovery partners, encrypt for recovery on a different iSeries system, and help protect you against accidental loss of your keys. Additional Decryption Keys are specified on the Alliance PGP Encrypt (PGP ENCRYPT) command. You can specify one primary key for encryption, and up to three additional decryption keys. Specifying an additional decryption key allows other key holders to decrypt the file.

If you are sending an encrypted tape to a vendor or customer, you can use your own PGP key as the additional decryption key. This allows you to decrypt the file in the event the tape is lost, and you need to recover the data. It also provides you with audit compliance and privacy notification protection as you can always recover the data and prove its contents.

If you are encrypting a tape for backup and archival, you may wish to encrypt the backup to your own PGP key, and use additional decryption keys of your back up iSeries platform or Business Recovery partner. You will always be able to decrypt files that are encrypted to your own key. In the event of a catastrophic loss of your production iSeries platform you can also restore and decrypt the save on your backup iSeries platform. If you have contracted for Business Recovery services with an outside vendor, they can decrypt the save using the PGP key on their own system.

The use of Additional Decryption Keys provides the most secure and recoverable method of protecting trading partner and backup tapes. It also provides you with the audit capabilities that are important for Sarbanes-Oxley compliance and Privacy Notification protection.

Alliance PGP encryption product considerations

The Alliance PGP encryption product has been used for tape encryption since its first release in 2001. Enhancements to the product have been made recently to make the use of the encryption and decryption routines easier with tape media. Existing Alliance customers can upgrade to version 3.61 or later to receive these enhancements. The examples below make use of these enhancements. There is no charge to receive these enhancements for Alliance customers with current maintenance contracts.

Example 1 - Encrypting a database file for tape transfer to a trading partner

This example shows the steps required to prepare a file in Excel CSV format, encrypt it with Alliance PGP, and copy it to tape for a trading partner. The name of the file is ORDERS in the library PRODLIB.

Step 1 – Convert the file to Excel CSV format.

The Alliance CrossData/400 product can be used to convert the file to CSV format. You can use interactive option on the Data Conversion menu to convert the file, or you can use one of the CrossData/400 API programs. From option 1 (Convert Database file to CSV) you would use these options:

File name	ORDERS
Library	PRODLIB
Member	ORDERS
Convert to file	ORDERSCSV
Library	PRODLIB
Member	ORDERSCSV

Step 2 – Encrypt the file using Alliance PGP.

In this example we will encrypt the file with Alliance PGP using the ASCII Armor option, convert the data to the ASCII character set, and copy the encrypted file to file ORDERSENC in library PRODLIB:

```
PGPENCRYPT IFSFILE('/tmp/Orders.csv')
  USERID('PartnerID')
  ARMOR(*YES)
  CPYOPT(*YES)
  FILE(ORDERSCSV/PRODLIB/ORDERSCSV)
  TRIM(*YES)
  CPYTOFILE(*YES)
  TOFILE(ORDERSENC/PRODLIB/ORDERSENC)
```

The result of this step is the creation of a file named ORDERSENC in library PRODLIB with the encrypted file. The data is in the ASCII character set with a fixed record length. The file is now ready to copy to the tape.

Step 3 – Copy the file to tape.

It is assumed that you have initialized the tape in your iSeries tape device. You can now use the Copy To Tape (CPYTOTAP) command to transfer the file to tape:

```
CPYTOTAP FROMFILE(PRODLIB/ORDERSENC)
  TOFILE(QTAPE)
  TODEV(TAP01)
  TOREELS(*SL)
  TORCDLEN(*FROMFILE)
  TOENDOPT(*REWIND)
  TORCDBLK(*FB)
```

This completes the process of preparing an encrypted file to send to a trading partner.

Example 2 – Encrypting an IFS file for tape transfer to a trading partner

This example shows the steps required to encrypt a file in an IFS directory and copy it to tape for a trading partner. The file may have been copied to the IFS directory by an iSeries application, have been transferred to the IFS directory by FTP, be a mounted UNIX directory on a web server, or be a Windows Networking directory under the QNTC IFS directory. The name of the file is Orders.csv in the IFS directory /mydir.

Step 1 – Encrypt the file using Alliance PGP.

In this example we will encrypt the file with Alliance PGP using the ASCII Armor option, and copy the encrypted file to file ORDERSENC in library PRODLIB:

```
PGPENCRYPT IFSFILE('/tmp/Orders.csv')
  USERID('PartnerID')
  ARMOR(*YES)
  CPYTOFILE(*YES)
  TOFILE(ORDERSENC/PRODLIB/ORDERSENC)
```

The result of this step is the creation of a file named ORDERSENC in library PRODLIB with the encrypted file. The data is in the ASCII character set with a fixed record length. The file is now ready to copy to the tape.

Step 2 – Copy the file to tape.

It is assumed that you have an initialized the tape in your iSeries tape device. You can now use the Copy To Tape (CPYTOTAP) command to transfer the file to tape:

```
CPYTOTAP FROMFILE(PRODLIB/ORDERSENC)
  TOFILE(QTAPE)
  TODEV(TAP01)
  TOREELS(*SL)
  TORCDLEN(*FROMFILE)
  TOENDOPT(*REWIND)
  TORCDBLK(*FB)
```

This completes the process of preparing an encrypted file to send to a trading partner.

Example 3 – Encrypting saved objects for backup archival

In this example objects are saved to a save file in a library, encrypted, and then saved to a tape device.

Step 1 – Save objects to a save file.

The first step is to save objects to a save file. If your current save routine is saving objects directly to tape, you can change it to save the objects to a save file instead. Create the save file, change the DEV parameter on the save command to *SAVF, and specify the save file. In this example all files that start with the characters ORD are saved to a save file named MYSAVF in library QGPL:

```
SAVOBJ
  OBJ(ORD*)
  LIB(PRODLIB)
  DEV(*SAVF)
  OBJTYPE(*FILE)
  SAVF(QGPL/MYSAVF)
```

Step 2 – Encrypt the save file and copy to a new file in a library.

In this step you use the Alliance PGP encryption command to encrypt the file using the ASCII Armor option, and copy the encrypted file to a library file to be saved. You can use the PGPENCRYPT command like this:

```
PGPENCRYPT IFSFILE('/tmp/mysavf.savf')
  USERID('MyID')
  ARMOR(*YES)
  CPYOPT(*YES)
  FILE(QGPL/MYSAVF)
  CONVERT(*NO)
  ENDLINFMT(*FIXED)
  TRIM(*NO)
  ADDUSERID('BackupUser')
  CPYTOFILE(*YES)
  TOFILE(MYSAVFENC/QGPL/MYSAVFENC)
```

Note that you can use your own PGP key to encrypt the file. This ensures that only you can decrypt the file on a restore operation. Also, since Alliance PGP supports the use of Additional Decryption Keys (ADK), you can encrypt the file for other recipients such as a Business Recovery vendor, etc. In this example the addition decryption key "BackupUser" is specified on the encryption command.

The result of this operation is the creation of the file MYSAVFENC in library QGPL with the encrypted contents of the original save file MYSAVF in library QGPL. The file MYSAVFENC is now ready for save to tape.

Step 3 – Save the encrypted file to tape.

You can now save the encrypted file to tape using the Save Objects (SAVOBJ) command:

```
SAVOBJ OBJ(MYSAVFENC)  
LIB(QGPL)  
DEV(TAP01)  
OBJTYPE(*FILE)
```

This completes the steps for saving selected objects to a save file and saving the resulting encrypted file to tape. See the examples below on how to restore this type of file and convert it back to save file format.

Example 4 – Encrypting a library for tape archival

In this example a library is saved to a save file encrypted, and then saved to a tape device.

Step 1 – Save the library to a save file.

The first step is to save the library to a save file. If your current save routine is saving the library directly to tape, you can change it to save the library to a save file instead. Create the save file, change the DEV parameter on the save command to *SAVF, and specify the save file. In this example the library PRODLIB is saved to a save file named MYSAVF in library QGPL:

```
SAVLIB
LIB(PRODLIB)
DEV(*SAVF)
SAVF(QGPL/MYSAVF)
```

Step 2 – Encrypt the save file and copy to a new file in a library.

In this step you use the Alliance PGP encryption command to encrypt the file using the ASCII Armor option, and copy the encrypted file to a library file to be saved. You can use the PGPENCRYPT command like this:

```
PGPENCRYPT IFSFILE('/tmp/mysavf.savf')
  USERID('MyID')
  ARMOR(*YES)
  CPYOPT(*YES)
  FILE(QGPL/MYSAVF)
  CONVERT(*NO)
  ENDLINFMT(*FIXED)
  TRIM(*NO)
  ADDUSERID('BackupUser')
  CPYTOFILE(*YES)
  TOFILE(MYSAVFENC/QGPL/MYSAVFENC)
```

Note that you can use your own PGP key to encrypt the file. This ensures that only you can decrypt the file on a restore operation. Also, since Alliance PGP supports the use of Additional Decryption Keys (ADK), you can encrypt the file for other recipients such as a Business Recovery vendor, etc. In this example the addition decryption key "BackupUser" is specified on the encryption command.

The result of this operation is the creation of the file MYSAVFENC in library QGPL with the encrypted contents of the original save file MYSAVF in library QGPL. The file MYSAVFENC is now ready for save to tape.

Step 3 – Save the encrypted file to tape.

You can now save the encrypted file to tape using the Save Objects (SAVOBJ) command:

```
SAVOBJ OBJ(MYSAVFENC)
LIB(QGPL)
DEV(TAP01)
OBJTYPE(*FILE)
```

Note that if you have several libraries to save you can accumulate the encrypted files in one library, and save the entire library to tape.

This completes the steps for saving a library to a save file and saving the resulting encrypted file to tape. See the examples below on how to restore this type of file and convert it back to save file format.

Example 5 – Restoring and decrypting a file from a trading partner

This example shows one way you can process an encrypted file you receive from a trading partner. This example assumes that the trading partner encrypted the file using PGP with the ASCII Armor option, and copied the encrypted file to tape with fixed record lengths.

Step 1 – Create a file to receive the encrypted file from tape

This step creates an internally described flat file to receive the contents of the encrypted file from tape:

```
CRTPF FILE(QGPL/RSTFILE)
RCDLEN(64)
SIZE(*NOMAX)
```

Note that you should ask your trading partner for the actual record length. Some ASCII Armored files have a record length of 64 bytes, some have record lengths of 65 bytes. Your trading partner can tell you the record length they used for the save to tape. You can also use the Dump Tape (DMPTAP) command to view information about the tape file.

Step 2 – Copy the encrypted file from tape to your flat file

You can now use the Copy From Tape (CPYFRMTAP) command to copy the encrypted file from tape to your flat file. This is an example of how that command might be used:

```
CPYFRMTAP FROMFILE(QTAPE)
TOFILE(QGPL/RSTFILE)
TOMBR(*FIRST)
FROMDEV(TAP01)
FROMREELS(*SL)
FROMENDOFT(*REWIND)
MBROPT(*REPLACE)
```

Note that most PGP encrypted files are in the ASCII character set. If you display the file you copied from tape the data may look like “garbage”. That is, the data may not be viewable using DFU or the Display Physical File Member (DSPPFM) command. This would be a normal condition.

Step 3 – Decrypt the file

You can now use the Alliance PGP Decrypt (PGPDECRYPT) command to decrypt the file. The result will be a decrypted file in the IFS directory system. You can use the command like this:

```
PGPDECRYPT IFSFILE('/tmp/myfile.txt.asc')
PASSWORD('My pass phrase')
CPYFRMFIL(*YES)
FROMFILE(RSTFILE/QGPL/RSTFILE)
```

Step 4 – Use the decrypted file

In most cases you will want to process the decrypted file into your application database. You can use the Alliance Directory Scan application to automatically pick up the file, copy it to a library, call your application program to process the data, and make an archive copy of the file. Alternatively, you can use the iSeries command Copy From Stream File (CPYFRMSTMF) to copy the data to a library for processing. If the data will be used by another system you can leave the decrypted file in a Windows Networking directory, or use Alliance automatic FTP functions to transfer the file to another server.

This completes the example showing how to use Alliance tape encryption routines to process a file that has been encrypted by a trading partner and sent to you.

Example 6 – Restoring and decrypting a library from tape archival

This example shows how you might restore a library that you saved in encrypted format using Alliance PGP encryption. This example assumes that you saved a library to a save file and then encrypted it in ASCII Armor format (see example 4 above). The resulting encrypted file was copied to a file in a library, and then saved to tape using normal Save Object (SAVOBJ) or Save Library (SAVLIB) command.

Step 1 – Restore the encrypted file from tape to a library

In this step the encrypted save file is restored from tape to a library. The data on the tape is in Save-Restore format, but the file retrieved is an internally described flat file, not a save file:

```
RSTOBJ OBJ(MYSAVFENC)
SAVLIB(QGPL)
DEV(TAP01)
OBJTYPE(*FILE)
```

Step 2 – Create a save file to receive the decrypted contents of the decryption

In this step we create a save file to receive the contents of the decrypted file. This save file is referenced on the Alliance PGP decryption command in the next step

```
CRTSAVF FILE(QGPL/MYRESTORE)
```

Step 3 – Decrypt the file and place the contents in the save file

In this step we decrypt the file specifying the source of the data as the flat file that we restored from tape, and specifying the target of the decrypted data as the new save file:

```
PGPDECRYPT IFSFILE('/tmp/mysavf.savf.asc')
PASSWORD('My pass phrase')
CPYFRMFL(*YES)
FROMFILE(MYSAVFENC/QGPL/MYSAVFENC)
CPYTOSAVF(*YES)
SAVF(QGPL/MYRESTORE)
```

This operation will copy the encrypted data from the internally described flat file to an IFS file named mysavf.savf.asc in the /tmp IFS directory, decrypt the file, and copy the contents of the decrypted file to the save file. You can display the contents of the save file using the Display Save File (DSPSAVF) command when the decryption is complete.

Step 4 – Restore the library

You can now restore the library using the Restore Library (RSTLIB) command. If the objects in the save file were saved using the Save Object (SAVOBJ) or Save (SAV) command, you can use the appropriate restore command to restore the objects:

```
RSTLIB
  SAVLIB (PRODLIB)
  DEV (*SAVF)
  SAVF (QGPL/MYRESTORE)
```

This completes the example of restoring an encrypted save file from a tape.