SafeNet KeySecure and PKWare

Integration Guide

THE
DATA
PROTECTION
COMPANY

Document Information

Document Part Number	007-012922-001 (Rev A)
Release Date	February 2015

Trademarks

Limitations

This document does not include the steps to set up the third-party software. The steps given in this document must be modified accordingly.

Disclaimer

The foregoing integration was performed and tested only with the specific versions of equipment and software and only in the configuration indicated. If your setup matches exactly, you should expect no trouble, and Customer Support can assist with any missteps. If your setup differs, then the foregoing is merely a template and you will need to adjust the instructions to fit your situation. Customer Support will attempt to assist, but cannot guarantee success in setups that we have not tested.

SafeNet makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, SafeNet reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon SafeNet to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

SafeNet invites constructive comments on the contents of this document. These comments, together with your personal and/or company details, should be sent to the address or email below.

Contact Method	Contact Information
Mail	SafeNet, Inc. 4690 Millennium Drive Belcamp, Maryland 21017, USA
Email	TechPubs@safenet-inc.com

Support Contacts

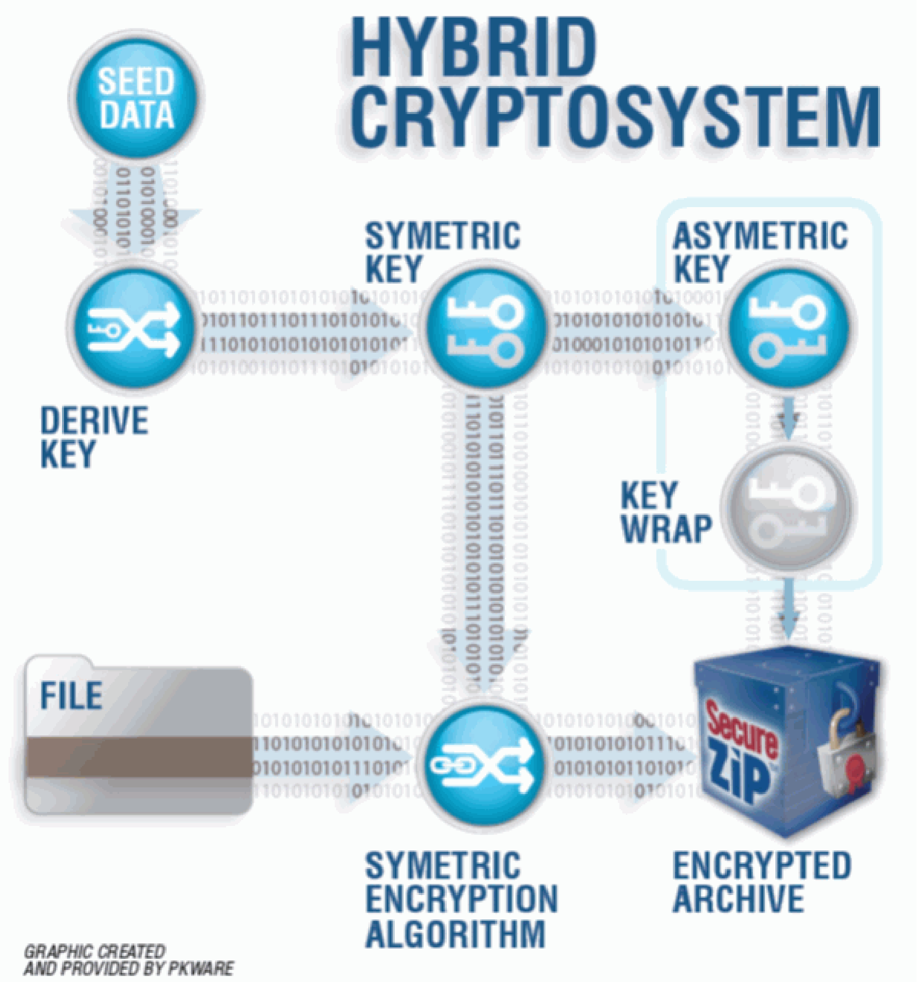
If you encounter a problem while installing, registering or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, contact your supplier or SafeNet Customer Support. SafeNet Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between SafeNet and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Contact Method	Contact Information	
Address	SafeNet, Inc. 4690 Millennium Drive Belcamp, Maryland 21017 USA	
Phone	United States	1-800-545-6608
	International	1-410-931-7520

Contents

CHAPTER 1
Introduction to PKWare SecureZip

The hybrid crypto solution offers a blend of the two different encryption key approaches, gaining the best benefits of each without the disadvantages of either. It automatically generates a random and complex symmetric session key to encrypt the target data, creating an encrypted payload. Hybrid systems then use the asymmetric public key of a public/private key pair to encrypt the symmetric key (see figure below). They apply the computationally-intensive asymmetric encryption to only the small symmetric key which is used to encrypt the larger target data payload. As a consequence, it consumes fewer resources while providing fast, effective encryption. Users never actually see or interact with the symmetric session key used to encrypt the target data. Instead, they interact with the external asymmetric keys for encrypting with the public key and decrypting with the private key. SecureZIP is implemented as a hybrid cryptosystem, strongly encrypting the symmetric private key with a public key.



SecureZIP uses random seed data to create the symmetric key used to encrypt the target data. This seed data (also known as ‘salt’ in some cryptographic writing) is generated by software-based cryptographic strength pseudo-random number generation (PRNG). The PRNG source varies by the operating system on which SecureZIP is deployed, but in all cases, the degree of randomness complies with the ANSI X9.62 - 1998 – Annex A.4. A given seed instance is generated using the PRNG source and other random system data including system time, process timing data and memory, process states and other random system parameters. Consequently, the symmetric key used to actually encrypt the target data achieves practically perfect randomness, and overcomes the weaknesses of human generated passphrases.

The derived ‘session’ symmetric key is used with the symmetric encryption algorithm to encrypt the target data that is then placed in the ZIP archive. The session symmetric key is then encrypted using the public key of the public/private key associated with the individual or individuals for whom the data is encrypted and that ‘wrapped’ symmetric key is placed with the archive as well. Encrypting the symmetric session key with the asymmetric key ensures that decryption has the operational efficiency of the former while protecting access to the data with the very high entropy of the latter.

CHAPTER 2
Integration Overview with SafeNet KeySecure

SafeNet provides a secure key storage device.

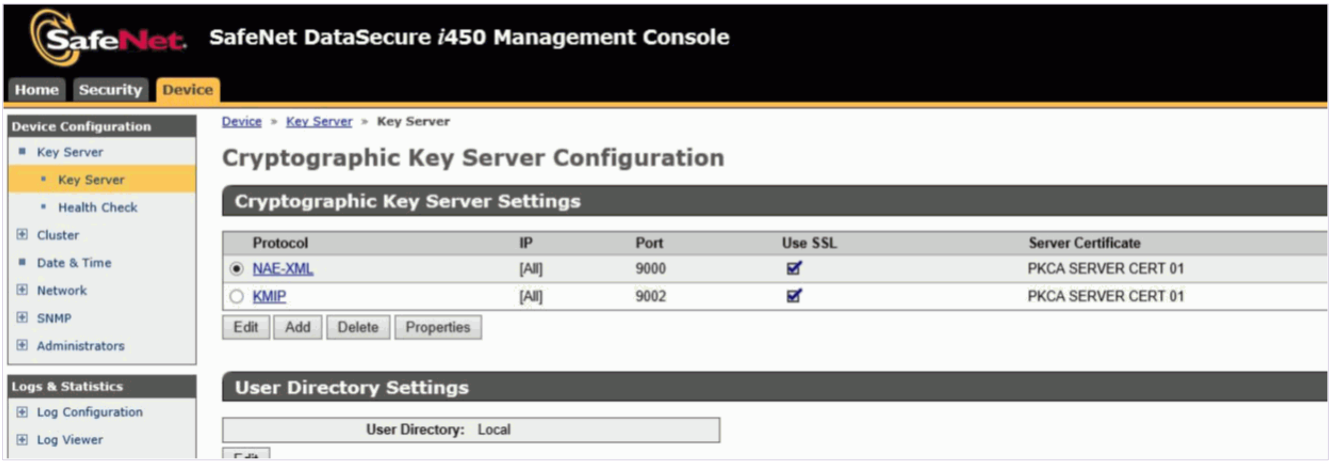
- Supported Key (Object) Types
- Password (stored as Secret Data)
- Public/Private RSA key pair (stored as Asymmetric Data)
- Key pair only (not suitable for signing)
- Key pair with Certificate (can be used for signing)

Interface

- SafeNet provides a web interface for all management functions
- Two main management functions:
  - Device (appliance configuration)
  - Security (keys, users, groups)

Device

- Device configuration requires two interfaces running on separate ports:
  - NAE (for connect/disconnect and all non-KMIP operations)
  - KMIP



Cryptographic Key Server Configuration

Cryptographic Key Server Settings

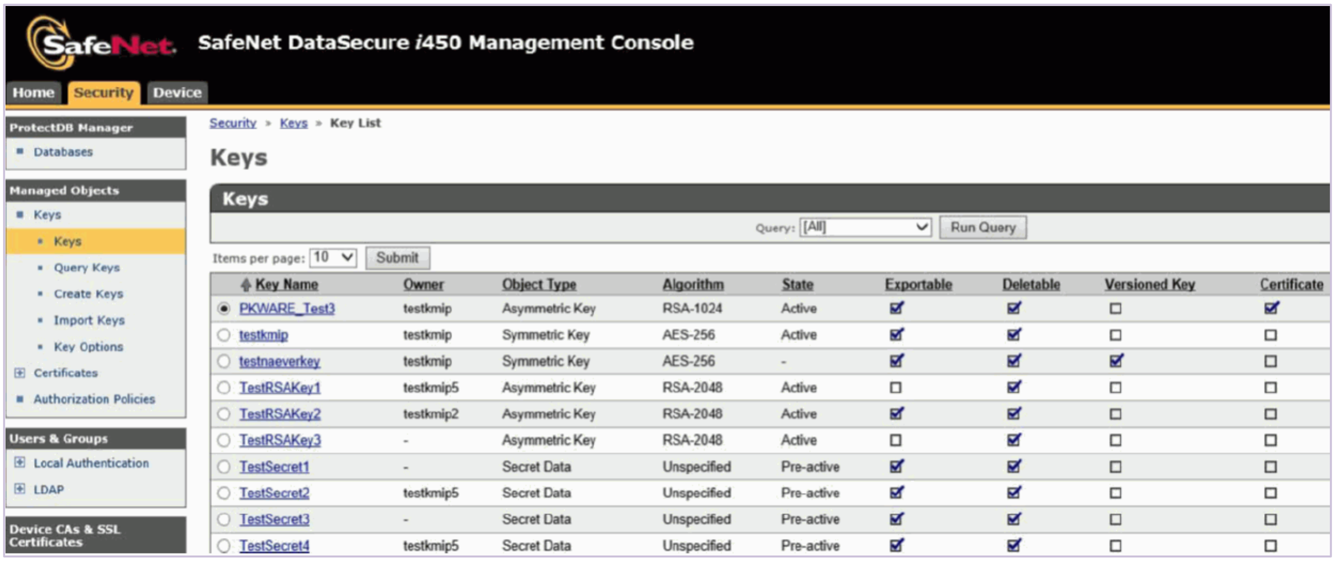
	Protocol	IP	Port	Use SSL	Server Certificate
●	NAE-XML	[All]	9000	☑	PKCA SERVER CERT 01
○	KMIP	[All]	9002	☑	PKCA SERVER CERT 01

User Directory Settings

User Directory: Local

Security

- Keys, users (key owners), and groups



Security » Keys » Key List

Keys

Keys

Query: [All]

Items per page: 10

	Key Name	Owner	Object Type	Algorithm	State	Exportable	Deletable	Versioned Key	Certificate
●	PKWARE_Test3	testkmip	Asymmetric Key	RSA-1024	Active	☑	☑	□	☑
○	testkmip	testkmip	Symmetric Key	AES-256	Active	☑	☑	□	□
○	testnaeverkey	testkmip	Symmetric Key	AES-256	-	☑	☑	☑	□
○	TestRSAKey1	testkmip5	Asymmetric Key	RSA-2048	Active	□	☑	□	□
○	TestRSAKey2	testkmip2	Asymmetric Key	RSA-2048	Active	☑	☑	□	□
○	TestRSAKey3	-	Asymmetric Key	RSA-2048	Active	□	☑	□	□
○	TestSecret1	-	Secret Data	Unspecified	Pre-active	☑	☑	□	□
○	TestSecret2	testkmip5	Secret Data	Unspecified	Pre-active	☑	☑	□	□
○	TestSecret3	-	Secret Data	Unspecified	Pre-active	☑	☑	□	□
○	TestSecret4	testkmip5	Secret Data	Unspecified	Pre-active	☑	☑	□	□

Keys

- Keys are identified/accessed using:
  - Key Name (unique label)
  - Unique ID (i.e. 278A69FD90DF83B7199D3EE3033877712D6EF619F257D2AC07BB261A7A104FB3)
  - Attributes (KMIP attributes)
- Keys can be associated to users:
  - as global keys – any authenticated user can access/use key (keys without “owners” are global).
  - as owned keys – only owner and associated groups can access/use key.

API’s

- Two API’s are required for interoperating with KeySecure:
  - NAE – SafeNet proprietary key and encryption interface
  - KMIP – partial implementation of KMIP protocol
- C/C++ and Java languages supported.
- Requires including SafeNet ingicapi.dll with SecureZIP.
- KMIP requires use of NAE API for functions not provided with KMIP.
  - Connect/Disconnect
  - Encrypt/Decrypt

  - Retrieve exportable public/private keys

Client Access

- Client applications communicate over configured ports (NAE and KMIP)
- Connection settings must be defined using SafeNet “properties file” passed to client application
  - Defines ports
  - Security settings
  - Logging

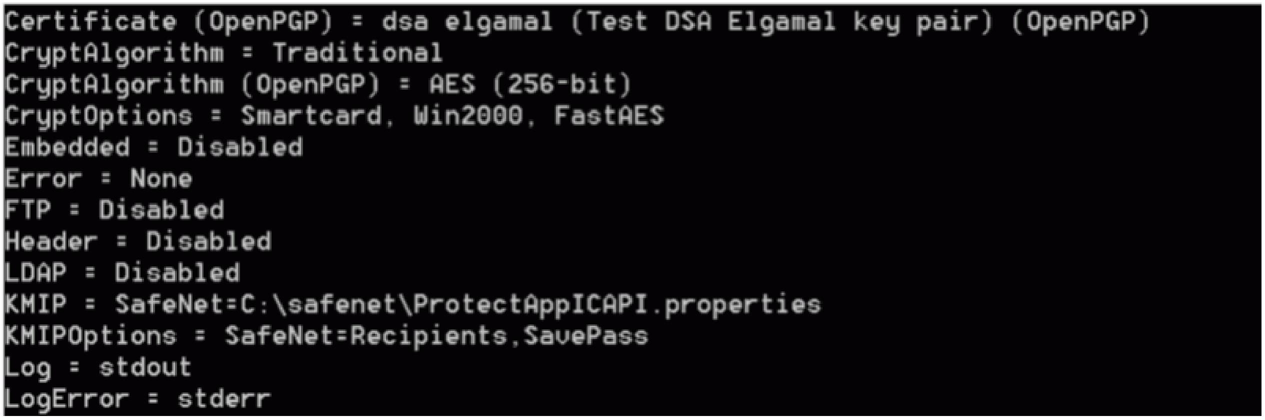
SecureZIP Server

- PKWARE’s Server product is modified to accept parameters to connect and communicate with SafeNet device.
  - *kmip* sets vendor and location of vendor configuration file (properties file)
  - *kmipoptions* sets vendor features to be used
- Locate recipients using KMIP.
- Store encryption passphrases using KMIP.
- KMIP options appear in SecureZIP only if SafeNet DLL is present.
- After configuration, SecureZIP operates using existing passphrase and recipient options Key Management.
- SecureZIP Server does not “manage” keys and SafeNet Management interface is required to create, import, expire, delete, and set owner and group associations for all keys.
- SecureZIP will “store” passphrases onto the KeySecure if configured for this operation and users with access to the device can use passphrase decryption by “key name”.

Sample Commands – Configure

- Configure SecureZIP for SafeNet key access:

```
pkzipc –config –kmip=SafeNet=C:\SafeNet\ProtectAPPICAPI.properties
pkzipc –config –kmipoptions=SafeNet=recipients,savepass
```

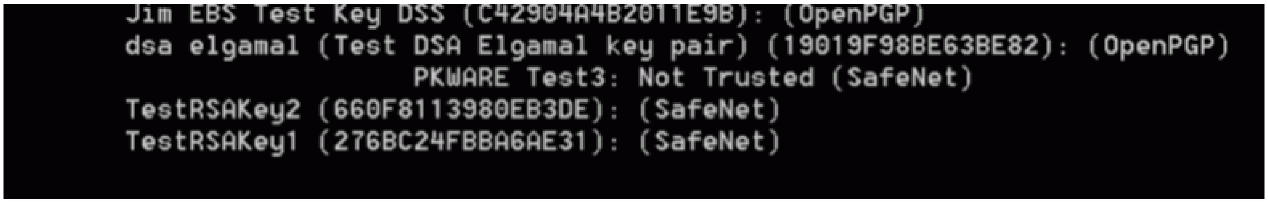


```
Certificate (OpenPGP) = dsa elgamal (Test DSA Elgamal key pair) (OpenPGP)
CryptAlgorithm = Traditional
CryptAlgorithm (OpenPGP) = AES (256-bit)
CryptOptions = Smartcard, Win2000, FastAES
Embedded = Disabled
Error = None
FTP = Disabled
Header = Disabled
LDAP = Disabled
KMIP = SafeNet=C:\safenet\ProtectAppICAPI.properties
KMIPOptions = SafeNet=Recipients,SavePass
Log = stdout
LogError = stderr
```

Sample Commands – List

- List available SafeNet key pairs:

  `pkzipc –listcert`



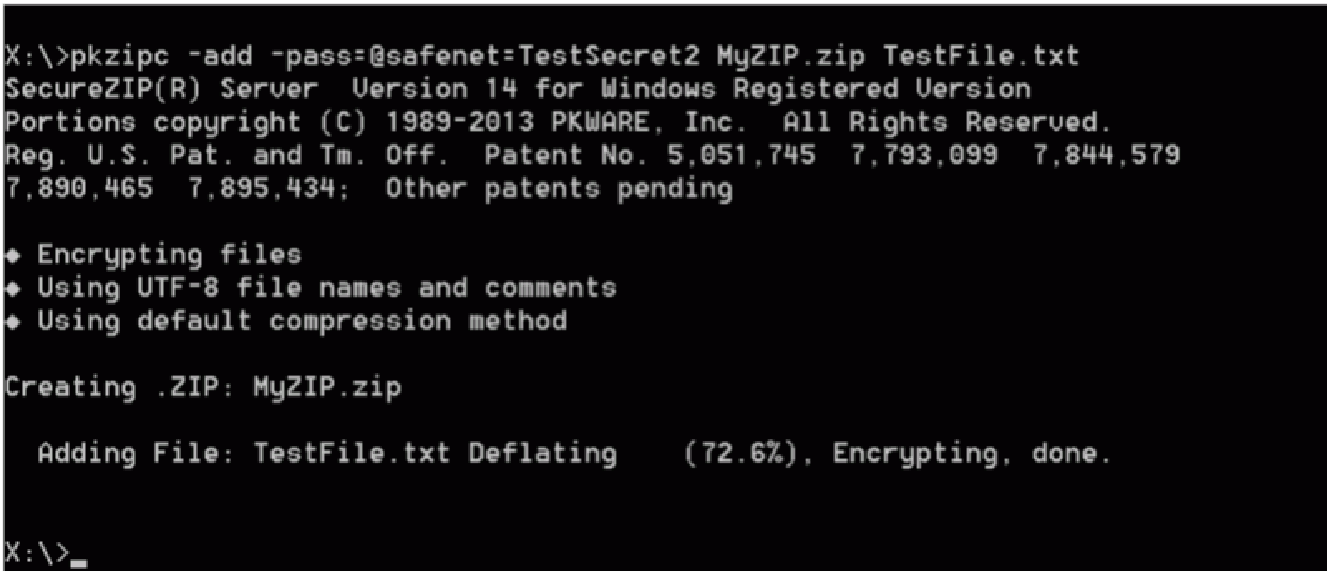
```
        Jim EBS Test Key DSS (C42904A4B2011E9B): (OpenPGP)
        dsa elgamal (Test DSA Elgamal key pair) (19019F98BE63BE82): (OpenPGP)
                          PKWARE Test3: Not Trusted (SafeNet)
        TestRSAKey2 (660F8113980EB3DE): (SafeNet)
        TestRSAKey1 (276BC24FBBA6AE31): (SafeNet)
```

- SafeNet key pairs appear with the name of the key provider
- “Stored” passphrases do not appear in listing
- SafeNet “key name” label appears for key pairs not having a certificate
- Certificate common name appears for key pairs within a certificate Object

Sample Commands – Passphrase

- Encrypt using an existing SafeNet “stored” passphrase:

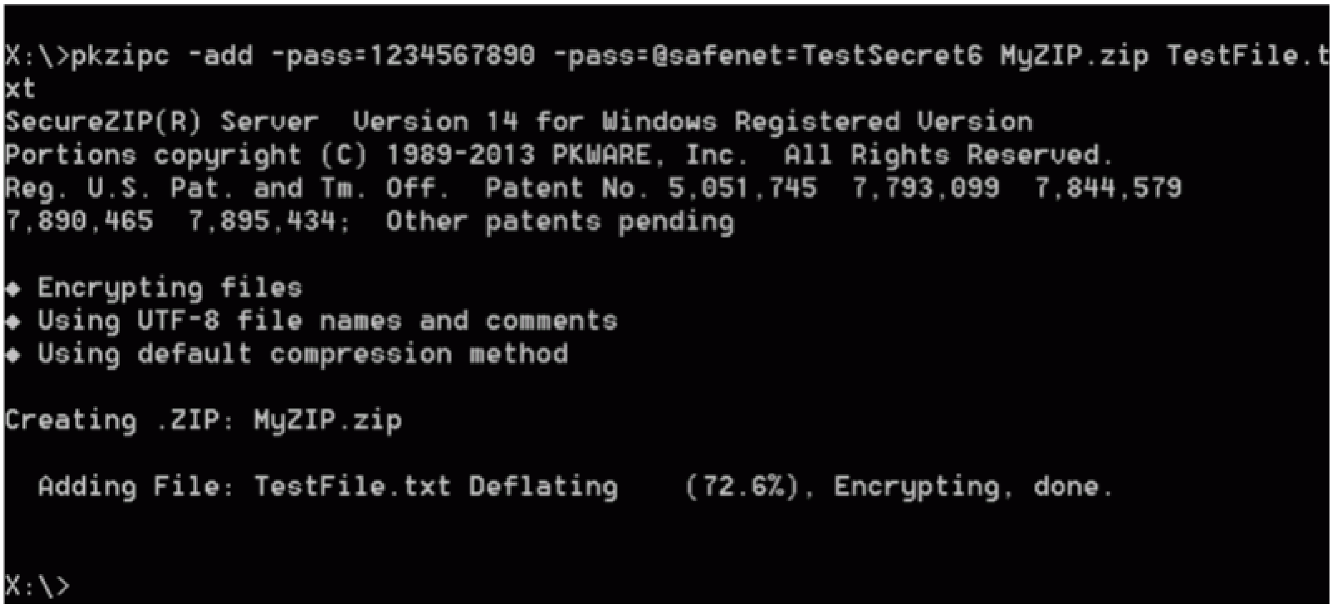
  `pkzipc –add –pass=@SafeNet=TestSecret2 MyZIP.zip TestFile.txt`



```
X:\>pkzipc -add -pass=@safenet=TestSecret2 MyZIP.zip TestFile.txt
SecureZIP(R) Server  Version 14 for Windows Registered Version
Portions copyright (C) 1989-2013 PKWARE, Inc.  All Rights Reserved.
Reg. U.S. Pat. and Tm. Off.  Patent No. 5,051,745  7,793,099  7,844,579
7,890,465  7,895,434;  Other patents pending

◆ Encrypting files
◆ Using UTF-8 file names and comments
◆ Using default compression method

Creating .ZIP: MyZIP.zip

  Adding File: TestFile.txt Deflating    (72.6%), Encrypting, done.


X:\>_
```

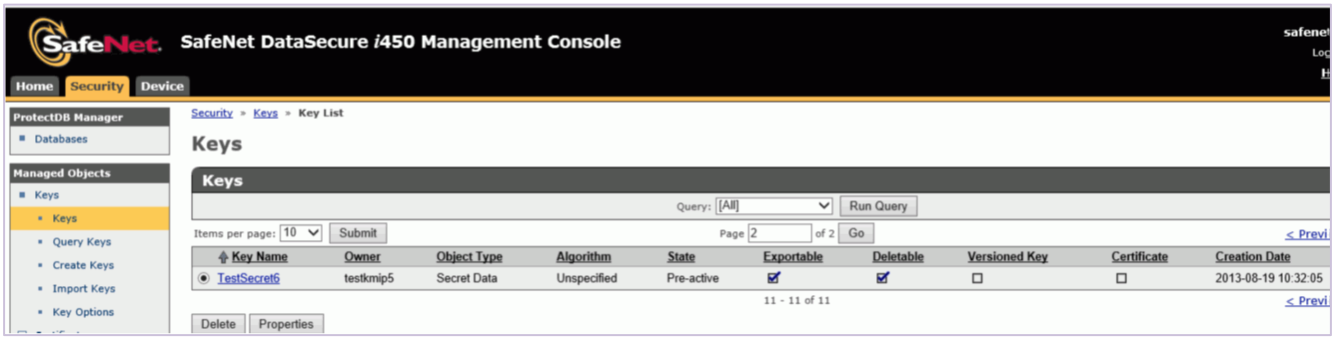
- Encrypt and store a “known” passphrase:

  ```
  pkzipc –add –pass=1234567890 –pass=@SafeNet=TestSecret6 MyZIP.zip
  TestFile.txt
  ```



```
X:\>pkzipc -add -pass=1234567890 -pass=@safenet=TestSecret6 MyZIP.zip TestFile.t
xt
SecureZIP(R) Server  Version 14 for Windows Registered Version
Portions copyright (C) 1989-2013 PKWARE, Inc.  All Rights Reserved.
Reg. U.S. Pat. and Tm. Off.  Patent No. 5,051,745  7,793,099  7,844,579
7,890,465  7,895,434;  Other patents pending

◆ Encrypting files
◆ Using UTF-8 file names and comments
◆ Using default compression method

Creating .ZIP: MyZIP.zip

  Adding File: TestFile.txt Deflating    (72.6%), Encrypting, done.


X:\>
```

A new “secret data” object is placed onto the SafeNet device by SecureZIP to store the user defined Passphrase.

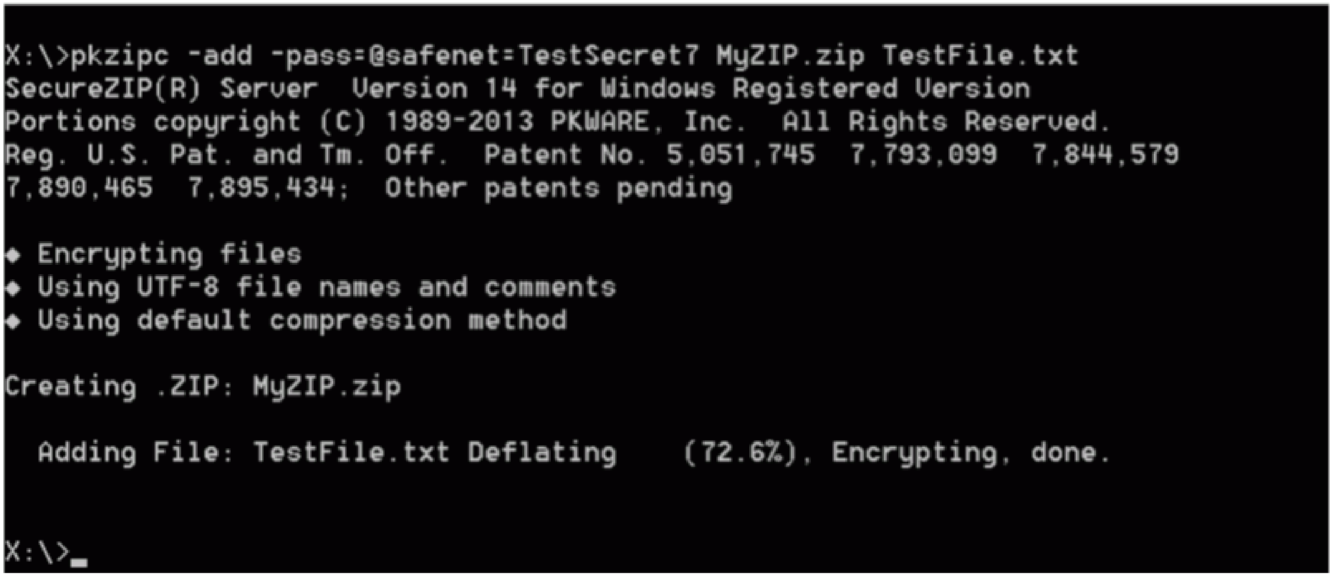


	Key Name	Owner	Object Type	Algorithm	State	Exportable	Deletable	Versioned Key	Certificate	Creation Date
●	TestSecret6	testkmip5	Secret Data	Unspecified	Pre-active	☑	☑	□	□	2013-08-19 10:32:05

11 - 11 of 11

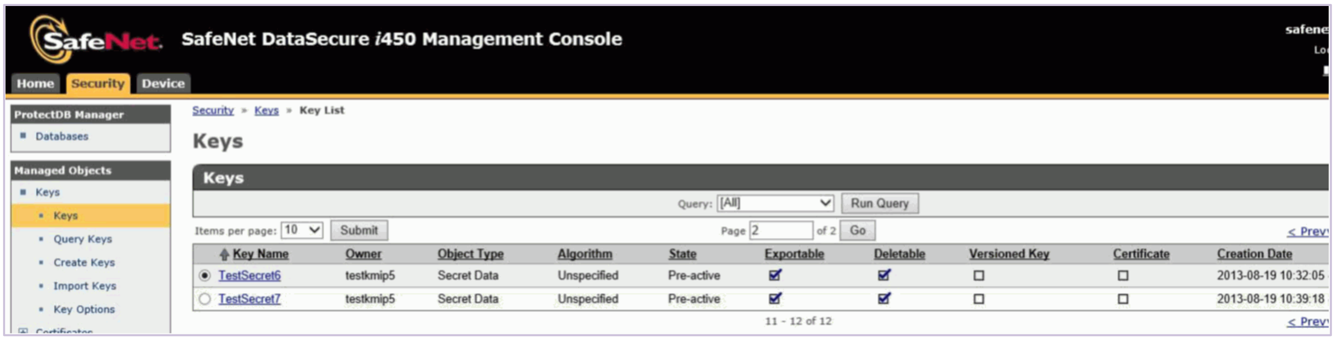
- Encrypt and store a random passphrase:

  `pkzipc –add –pass=@SafeNet=TestSecret7 MyZIP.zip TestFile.txt`



```
X:\>pkzipc -add -pass=@safenet=TestSecret7 MyZIP.zip TestFile.txt
SecureZIP(R) Server  Version 14 for Windows Registered Version
Portions copyright (C) 1989-2013 PKWARE, Inc.  All Rights Reserved.
Reg. U.S. Pat. and Tm. Off.  Patent No. 5,051,745  7,793,099  7,844,579
7,890,465  7,895,434;  Other patents pending

◆ Encrypting files
◆ Using UTF-8 file names and comments
◆ Using default compression method

Creating .ZIP: MyZIP.zip

  Adding File: TestFile.txt Deflating    (72.6%), Encrypting, done.


X:\>_
```

A new “secret data” object is placed onto the SafeNet device by SecureZIP to store the SecureZIP created random passphrase.



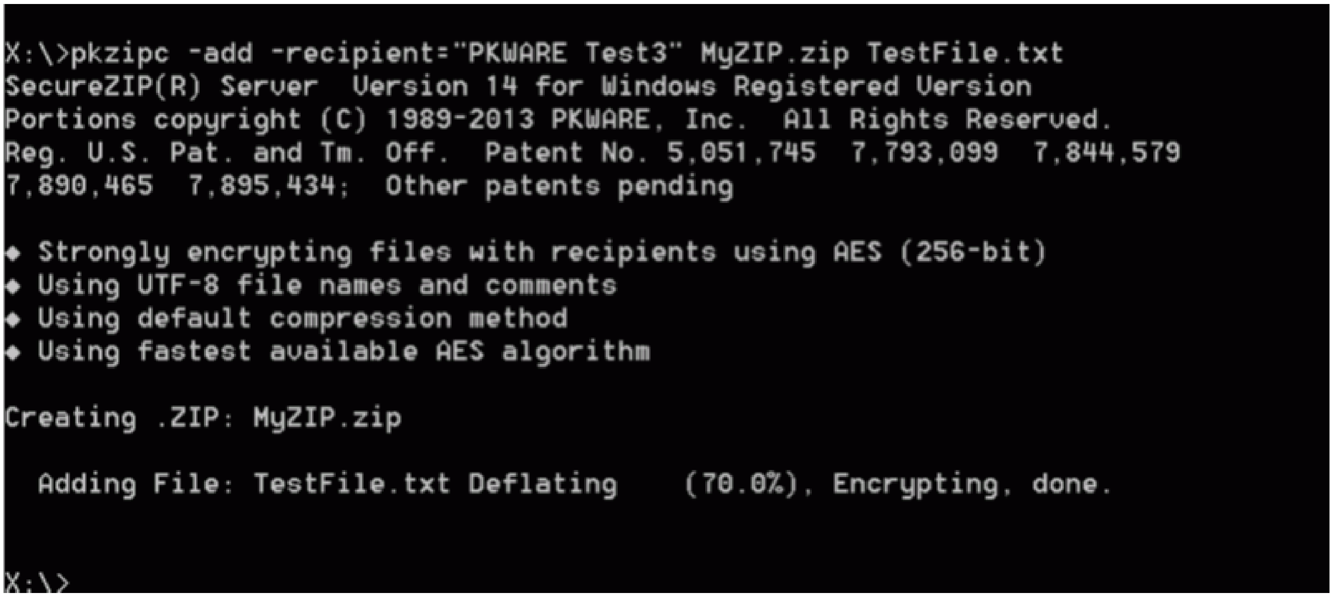
	Key Name	Owner	Object Type	Algorithm	State	Exportable	Deletable	Versioned Key	Certificate	Creation Date
●	TestSecret6	testkmip5	Secret Data	Unspecified	Pre-active	☑	☑	□	□	2013-08-19 10:32:05
○	TestSecret7	testkmip5	Secret Data	Unspecified	Pre-active	☑	☑	□	□	2013-08-19 10:39:18

11 - 12 of 12

Sample Commands – Certificate

- Encrypt using a certificate:

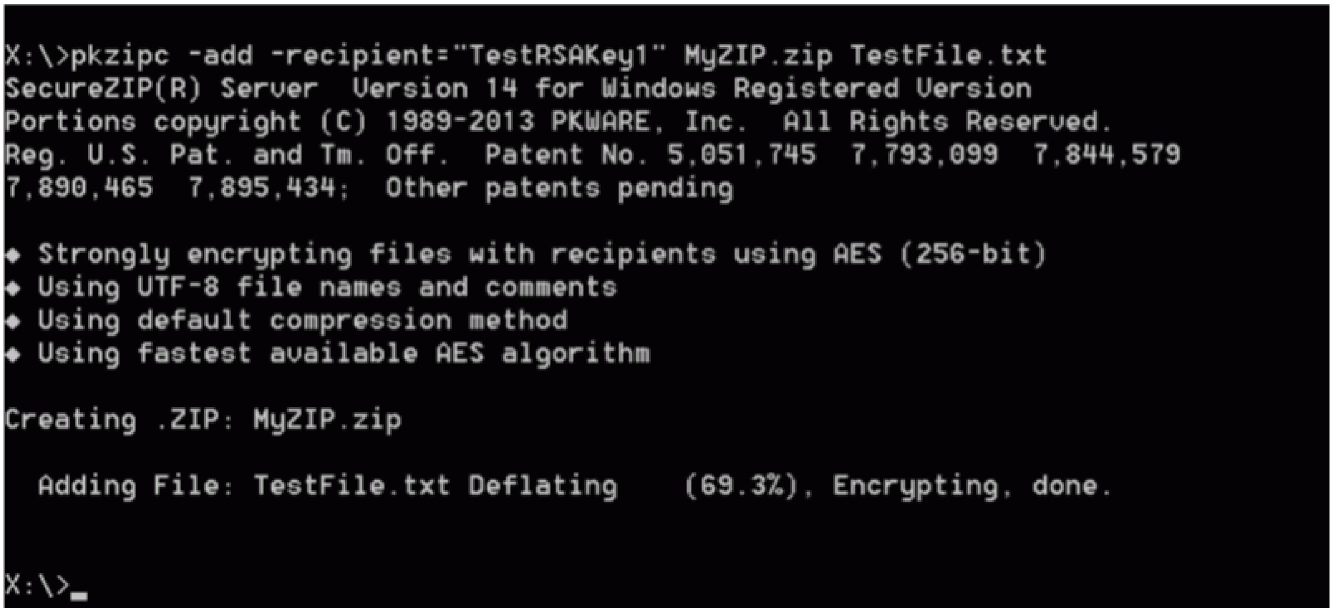
  `pkzipc –add –recipient=“PKWARE Test3” MyZIP.zip TestFile.txt`



```
X:\>pkzipc -add -recipient="PKWARE Test3" MyZIP.zip TestFile.txt
SecureZIP(R) Server  Version 14 for Windows Registered Version
Portions copyright (C) 1989-2013 PKWARE, Inc.  All Rights Reserved.
Reg. U.S. Pat. and Tm. Off.  Patent No. 5,051,745  7,793,099  7,844,579
7,890,465  7,895,434;  Other patents pending

◆ Strongly encrypting files with recipients using AES (256-bit)
◆ Using UTF-8 file names and comments
◆ Using default compression method
◆ Using fastest available AES algorithm

Creating .ZIP: MyZIP.zip

  Adding File: TestFile.txt Deflating    (70.0%), Encrypting, done.


X:\>
```

Sample Commands – Key Name (no certificate)

- Encrypt using a public key “key name” from a SafeNet Private Key Object:

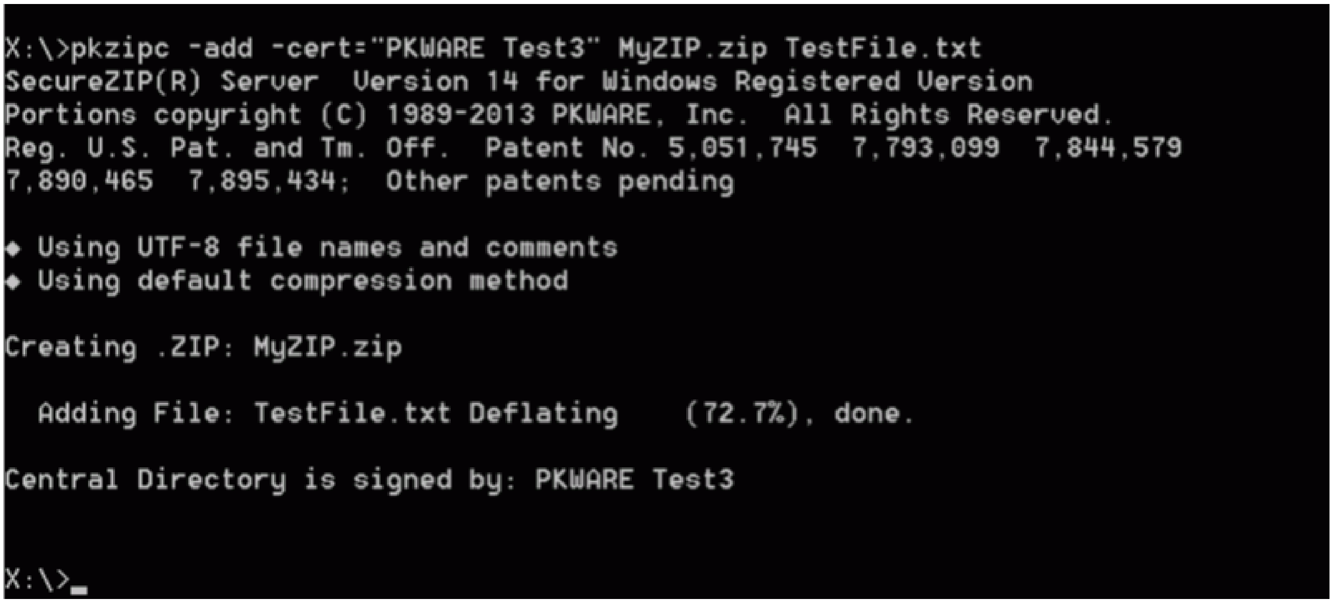


```
X:\>pkzipc -add -recipient="TestRSAKey1" MyZIP.zip TestFile.txt
SecureZIP(R) Server  Version 14 for Windows Registered Version
Portions copyright (C) 1989-2013 PKWARE, Inc.  All Rights Reserved.
Reg. U.S. Pat. and Tm. Off.  Patent No. 5,051,745  7,793,099  7,844,579
7,890,465  7,895,434;  Other patents pending

◆ Strongly encrypting files with recipients using AES (256-bit)
◆ Using UTF-8 file names and comments
◆ Using default compression method
◆ Using fastest available AES algorithm

Creating .ZIP: MyZIP.zip

  Adding File: TestFile.txt Deflating    (69.3%), Encrypting, done.


X:\>_
```

Sample Commands – Signing

- Sign a ZIP file using a certificate:

  `pkzipc –add –cert=“PKWARE Test3” MyZIP.zip TestFile.txt`



```
X:\>pkzipc -add -cert="PKWARE Test3" MyZIP.zip TestFile.txt
SecureZIP(R) Server  Version 14 for Windows Registered Version
Portions copyright (C) 1989-2013 PKWARE, Inc.  All Rights Reserved.
Reg. U.S. Pat. and Tm. Off.  Patent No. 5,051,745  7,793,099  7,844,579
7,890,465  7,895,434;  Other patents pending

◆ Using UTF-8 file names and comments
◆ Using default compression method

Creating .ZIP: MyZIP.zip

  Adding File: TestFile.txt Deflating    (72.7%), done.

Central Directory is signed by: PKWARE Test3


X:\>_
```

CHAPTER 3

Integration Benefits of SecureZIP with SafeNet KeySecure

The following are the benefits of SafeNet KeySecure integration with SecureZIP:

- Passphrase encryption is enhanced through new ability to “store” passphrases securely on the SafeNet device.
- Passphrase encryption is enhanced through automatic passphrase (random 240 characters) generation for “stored” passphrases.
- Certificate encryption is limited to “exportable” certificates only.
- Certificates can be selected by SafeNet “key name” in addition to Common name and Email address.
- OpenPGP keys cannot be stored or retrieved at this time.
- SafeNet “private key” objects do not include an X.509 certificate and can be used for encryption only (no signing).

Conclusion

PKWARE’s SecureZIP, configured appropriately with SafeNet KeySecure for the level of protection required, delivers data encryption to a standard expected to remain highly durable in the face of attack, even if such attacks are driven by massive parallel processing using the latest processors, whether CPU or GPU. It is the data protection application of choice by major enterprises, small office/home office users, and consumers, worldwide, due as much for its highly competent implementation of hybrid crypto system architecture as for its ease of use.