Sharing Secrets Using Encryption Facility

Eysha S. Powers
IBM Corporation

Tuesday, August 11, 2015: 6:00pm – 7:00pm
Session Number 17624

#SHAREorg

Cryptography is used in a variety of places…

- **Data in flight**
  - Virtual Private Networks (VPNs)
  - SSL/TLS connections (using public/private keys and certificates and symmetric encryption)
  - Messaging infrastructures (using SSL/TLS or shared secrets)
  - WS-Security and SOA
- **Data at rest**
  - File and folder encryption – including the use of intermediate devices
  - Removable media (tape) encryption
- **Transactional environments**
  - Industry specific – finance
  - Mandates highly trust-worthy cryptography
  - Smart ID cards, ePassports…
- **For sharing user credentials between organizations – the establishment of trust**
  - Via certificate exchanges
  - Federated Identity Management
  - Credential formats such as SAML, OpenID Connect…

Encryption Facility for z/OS

The Encryption Services feature supports encrypting and decrypting various file formats on z/OS. This can allow you to transfer them to remote sites within your enterprise, transfer them to partners and vendors, and archive them.

Encryption Facility for z/OS provides services for:

- Public-key based encryption
- Passphrase-based encryption
- Modification detection of encrypted data
- Compression of packaged data before encryption
- Importing and exporting of OpenPGP certificates
  - Binary or ASCII armor format
- Digital signatures of data

Encryption Facility can be run from the USS command line or from the JZOS Batch Launcher

Encryption Facility for z/OS: Components

Encryption Facility for z/OS consists of two priced optional features:

Encryption Services

For encryption and decryption of z/OS files and datasets. Uses OpenPGP or System z message format.

DFSMSdss Encryption

For encryption and decryption of z/OS dump datasets.

Independently, the Encryption Facility for z/OS Client is a no-cost, separately licensed program (which is offered as is, with no warranty) and is designed to enable the exchange of encrypted data:

Client

Encrypt non-z/OS files to be sent to z/OS. Decrypt z/OS files to be sent to non-z/OS platforms. Uses a System z message format.

OpenPGP Support

OpenPGP is a widely accepted, open standard for handling encryption of data files and messages. (RFC 4880)

OpenPGP allows customers to exchange an encrypted, compressed, and/or digitally signed file with business partners who have an installed OpenPGP client running on z/OS and other operating systems.

OpenPGP is:

- A file format for exchanging encrypted data.
  - It is not an encryption algorithm.
- Is standardized by the IETF and has been adopted as the format for encrypted files by a wide-range of open-source and commercial products
- Does not define how encryption keys should be managed
- Allows a wide range of choices for key exchange and trust model
- Provides additional flexibility and interoperability for tape exchanges with external business partners and vendors

OpenPGP Support for Encryption Facility

Provides at a minimum, support for all Mandatory/Must Do's identified in the OpenPGP standard (RFC 4880).

File Operations
- Encryption
  - Passphrase-based encryption
  - Public key encryption
- Compression
- Digital Signatures

Public Key Generation
- Generating RSA, DSA and Elgamal keys

Open PGP Certificates
- Importing and Exporting OpenPGP Certificates
- ASCII Armor for OpenPGP Certificates

- **Compression Algorithms**
  - ZLIB
    - Hardware-Based (zEDC)
    - Software-Based
  - ZIP
- **Symmetric Encryption Algorithms**
  - AES 128, 192 and 256 bits
  - Triple DES
  - Blowfish
- **Asymmetric Encryption Algorithms**
  - RSA
  - Elgamal
- **Digest / Hash Algorithms**
  - MD2, MD5
  - SHA1, SHA-256, SHA-384, SHA-512
- **Digital Signature Algorithms**
  - RSA with all supported hash algorithms above
  - DSA with SHA1

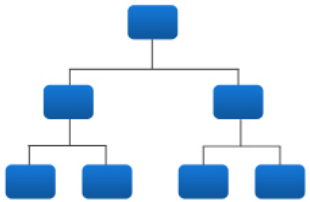
Using Certificates

Digital certificates contain a public key, information to identify who the key belongs to, and a digital signature to authenticate and bind together the public key with the identity information. Digital signatures are made by taking a hash of message, in this case the digital certificate, and then encrypting the hash value with the signer's private key.

Encryption Facility can use public keys to encrypt data. Public keys are exchanged between business partners using digital certificates. Only the signer has the private key that was used to create the signature. The business partners must have the associated public key in order to verify the signature.

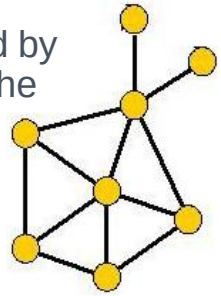
X.509 Certificates

- Use a hierarchical authentication model
- Each X.509 certificate contains one digital signature, either self signed or signed by a CA
- A root Certificate Authority (CA) is established and trusted as a self signed certificate
- Other certificates are signed by a CA within the hierarchy



Open PGP Certificates

- Use a decentralized authentication model
- Each OpenPGP certificate can be self signed and can contain multiple signatures from other keys
- OpenPGP sub-keys are all signed by the Primary key to bind together the Primary and sub-keys
- Encryption Facility for z/OS provides an option to sign your OpenPGP certificates with a CA.



Configuring Encryption Facility for z/OS

Encryption Facility leverages ICSF, RACF and/or JCE to:

1. Setup keys to be used for encryption
2. Call the appropriate library to encrypt data with a given key
3. Format the output encrypted data into a message that can be later decrypted

ICSF Only (JCECCA)

- Generate clear or secure RSA key pairs in the PKDS
- Use existing RSA key pairs in the PKDS

RACF Only (JCERACFKS)

- Use existing RSA key pairs connected to a RACF key ring

JCE Only (JCEKS)

- Generate clear or RSA key pairs with software based cryptography
- Use existing RSA key pairs in a Java keystore in Unix Systems Services (USS)

ICSF & RACF (JCECCARACFKS)

- Use existing RSA key pairs connected to a RACF key ring that reference private keys stored in the PKDS

EF OpenPGP Command Syntax

Encryption Facility for OpenPGP commands have the following syntax:

[***-homedir*** <path>] [***-options*** [<*arguments*>]] ***-commands*** [<*arguments*>]

homedir: indicates the path to the ibmef.config file which can be used to set default
configuration options.

options: the name of one or more configuration options. Options always start with “-” and may be followed by one or more arguments. These values override any values set in the configuration file.

commands: Name of one or more operational commands. Commands always start
with “-” and may be followed by one or more arguments.

Invoking Encryption Facility for z/OS

Encryption Facility can be invoked in the following two ways:

- USS using a command terminal or shell script
- JCL using the IBM JZOS Java Batch Launcher

The IBM JZOS Java Batch Launcher is a standard feature of the IBM's Java SDK and provides the ability to launch Java applications through JCL.

Using the JZOS Batch Launcher requires the following:

- JZOS Procedure in PROCLIB
- Shell script to configure environment variables
- JCL/Batch job that calls the procedure in PROCLIB

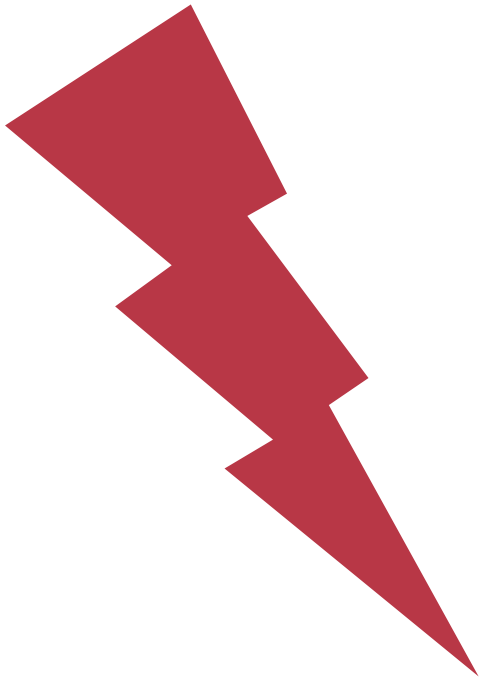
Sample JZOS files are provided as part of the Encryption Facility for z/OS package and can also be found under the references and sample section within the user guide.

Use of zIIP and zAAP Specialty Engines

zIIP and zAAP processors are specialty processors designed to lower overall computing cost by offloading certain types of workloads, such as Java workloads.

- If a zAAP or zIIP processor is available on the system, Encryption Facility OpenPGP jobs will be off-loaded to those processors.
- When called from Java using the JCE HW provider, ICSF code remains zIIP and zAAP eligible.

Java 8 Performance Improvements



The combined benefits of IBM Java 8 and z13 features – including Single Instruction Multiple Data (SIMD) vector engine, simultaneous multithreading (SMT) and improved CP Assist for Cryptographic Function (CPACF) – are providing up-to 2X improvement in throughput-per-core for security-enabled applications and up-to 50% improvement for other generic applications.

IBM Encryption Facility for z/OS is another Java application that leverages the new Java 8 CPACF exploitation. Encryption of text files and SVC dumps completed in half the elapsed time and one third the CPU time.

Encryption Facility for z/OS: Hands-On Lab

Additional Resources…

Encryption Facility for z/OS

- SA23-2229 IBM Encryption Facility for z/OS: Planning and Customizing
- SA23-2230 IBM Encryption Facility for z/OS: Using Encryption Facility for OpenPGP

Publications may be downloaded from the z/OS Internet Library

http://www-03.ibm.com/systems/z/os/zos/library/bkserv/v2r1pdf/#CSD

Encryption Facility for z/OS Website

http://www-03.ibm.com/systems/z/os/zos/tools/encryption_facility/

Thank You!

Feel free to connect…

- Email: eysha@us.ibm.com
- Twitter: http://www.twitter.com/EyshaShirrine
- LinkedIn: http://www.linkedin.com/in/eysha