Crypto Coding Lab: Welcome to the CIA!

Eysha S. Powers
Enterprise Cryptography
IBM Corporation

2015

IBM Systems Technical University

IBM z Systems • IBM Power Systems • IBM Storage

October 5–9 | Hilton Orlando, Florida

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Session Goals

Learn how to code ICSF crypto applications

- Identifying API Naming Conventions
- Using the Application Programmer's Guide
- Performing basic crypto operations
  - Key Generation
  - Key Distribution
  - Data Encrypt & Decrypt
  - Data Hashing
  - Digital Signature Generation & Verification

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Information Security Basics

- **Confidentiality:** Ensuring information is accessible only to those authorized to have access
- **Integrity:** Safeguarding the accuracy and completeness of information and processing methods
- **Availability:** Ensuring that authorized users have access to information and associated assets when required

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Integrated Cryptographic Services Facility (ICSF)

ICSF works with the hardware cryptographic features and the Security Server (RACF element) to provide secure, high-speed cryptographic services in the z/OS environment.

- ICSF provides the application programming interfaces by which applications request cryptographic services.
- ICSF is the default means by which the secure cryptographic features are loaded with master key values, allowing the hardware features to be used by applications.
- ICSF callable services and programs can be used to generate, maintain, and manage keys that are used in the cryptographic functions.

ICSF uses keys in cryptographic functions to

- Protect data
- Protect other keys
- Verify that messages were not altered
- Generate, protect and verify PINs
- Distribute keys
- Generate and verify signatures

ICSF Key Data Sets

CKDS

Cryptographic Key Data Set
- CCA Symmetric Keys
- AES and DES

PKDS

PKA Key Data Set
- CCA Asymmetric Keys
- RSA and ECC

TKDS

Token Key Data Set
- PKCS#11 Keys, Certificates
- All algorithms

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Common Cryptographic Architecture (CCA) for z/OS ICSF

IBM Common Cryptographic Architecture (CCA)

- IBM proprietary cryptographic application programmers interface (API) providing a broad range of cryptographic services including
  - standard cryptographic algorithms
  - financial services standards

z/OS ICSF Naming Conventions for CCA

CSNB* = CCA 31-bit Symmetric Key API
CSNE* = CCA 64-bit Symmetric Key API
CSND* = CCA 31-bit Asymmetric Key API
CSNF* = CCA 64-bit Asymmetric Key API

CCA Functions & Algorithms

Encrypt / Decrypt (AES, DES, DES3, RSA)
Sign / Verify (RSA, ECC)
MAC Generate / Verify (AES, DES, DES3)
HMAC Generate / Verify (HMAC)
Key Generate (AES, DES, DES3, HMAC)
Key Pair Generate (RSA, ECC)
Key Agreement (ECC, DH)
One Way Hash
Random Number Generate
Key Import / Export
TR-31 Block Import / Export
Financial Crypto
PIN Generate / Verify / Translate
PIN Encrypt
Diversified Key Generate
Derive Unique Key Per Transaction (DUKPT)
CVV Generate / Verify
Secure Messaging for Keys / Pins
… And Many More!

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

PKCS#11 Cryptographic Token Interface Standard for z/OS ICSF

PKCS #11 Cryptographic Architecture

- Originally published by RSA Laboratories, now maintained by OASIS
  - Defines a standard API for devices that hold cryptographic information and perform cryptographic functions
  - Enterprise PKCS#11 – EP11

z/OS ICSF Naming Convention for PKCS#11

CSFP* = PKCS#11 APIs

PKCS#11 Functions & Algorithms

Encrypt / Decrypt (AES, DES, TDES, RSA)
Sign / Verify (RSA, DSA, ECDSA)
HMAC Generate / Verify
Key Generate (DES, TDES, AES, Blowfish, RC4)
Key Pair Generate (RSA, DSA, EC)
Key Derivation
Domain Parameter Generation (DH)
One Way Hash
Random Number Generate
Wrap / Unwrap Key

Designed for portability and FIPS/Common Criteria certification

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Your Mission… Should you choose to accept it…

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Let's Begin!

Sender

1. Generate Keys
2. Encrypt Data
3. Hash the Cipher Text
4. Sign the Hash

Receiver

- Receive a public key
- Receive some cipher text
- Receive a signature

- Share the public key
- Share the cipher text
- Share the signature

1. Hash the Cipher Text
2. Verify the Signature
3. Decrypt the Data

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

But Where Do We Start?

- Which ICSF API should we use?
  - What functionality?
  - What architecture?
  - What algorithm?
  - What key type?

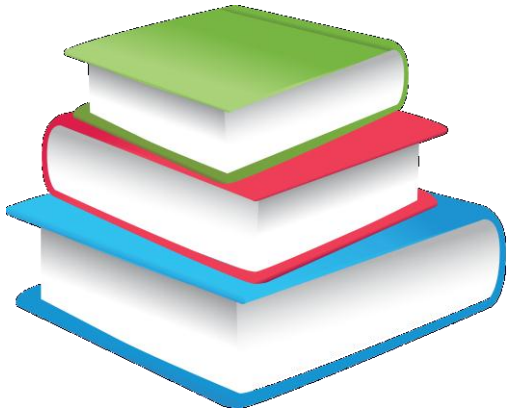
© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Cryptographic Services ICSF: Application Programmer's Guide

- Describes how to use the callable services provided by the Integrated Cryptographic Service Facility (ICSF).
- Intended for application programmers who:
  - Are responsible for writing application programs that use the security application programming interface (API) to access cryptographic functions.
  - Want to use ICSF callable services in high-level languages such as C, COBOL, FORTRAN, and PL/I, as well as in assembler.

http://www-01.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.csf/csf.htm

- Parameters
- Key Token Formats
- Return & Reason Codes



© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Step-By-Step: Encryption

- Generate a secret key
  - **SHARB01.ICSFLAB.REXX(KEYGEN1)**
- Generate a public-private key pair
  - **SHARB01.ICSFLAB.REXX(KEYGEN2)**
- Encrypt data with the secret key
  - **SHARB01.ICSFLAB.REXX(ENCRYPT)**
- Hash the encrypted data
  - **SHARB01.ICSFLAB.REXX(HASH)**
- Sign the hash with the private key
  - **SHARB01.ICSFLAB.REXX(SIGN)**

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Step-By-Step: Key Distribution

- Extract the public key from the private key
  - **SHARB01.ICSFLAB.REXX(KEYEXTRC)**

Choose a receiver's userid

- Encrypt the secret key with the receiver's public key
  - **SHARB01.ICSFLAB.REXX(KEYEXPOR)**

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Step-By-Step: Decryption

- Generate a hash of the cipher text
  - **SHARB01.ICSFLAB.REXX(HASH)**
  - **Manual Check:** Does the hash match the sender's hash?
- Verify the hash with the receiver's public key
  - **SHARB01.ICSFLAB.REXX(VERIFY)**
- Import the sender's secret key
  - **SHARB01.ICSFLAB.REXX(KEYIMPOR)**
- Decrypt the data with the sender's secret key
  - **SHARB01.ICSFLAB.REXX(DECRYPT)**

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

What You've Learned…

Cryptography enables you to protect your sensitive data:

- Encryption and Decryption to hide data
- Key Distribution to send encrypted data to other parties
- Digital Signatures to prove the originator of the data
- … and more

IBM z Systems provide both hardware and software cryptography to help you protect your IT assets.

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Additional Resources

IBM Crypto Education Community
https://www-304.ibm.com/connections/communities/community/crypto

IBM System z Development Blog
https://ibm.biz/zsystems_development

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Thank You!

Feel free to connect…

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

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Session Evaluations

YOUR OPINION MATTERS!

1 2 3 4

Submit four or more session evaluations by 5:30pm Wednesday to be eligible for drawings!

*Winners will be notified Thursday morning. Prizes must be picked up at registration desk, during operating hours, by the conclusion of the event.

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Continue growing your IBM skills

ibm.com/training

provides a comprehensive portfolio of skills and career accelerators that are designed to meet all your training needs.

If you can't find the **training that is right for you** with our Global Training Providers, we can help.

Contact IBM Training at dpmc@us.ibm.com

Global Skills Initiative

© Copyright IBM Corporation 2015. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.