z/OS V2.5 IBM Education Assistant

Solution Name:	KDS Support for Dilithium and TR-31 Key Blocks
Solution Element(s):	ICSF

July 2021

Agenda

- Trademarks
- Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Upgrade & Coexistence Considerations
- Installation & Configuration
- Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Objectives

- Discuss new function enhancements and benefits.

Overview

- Who (Audience)
    - Exploiters of quantum safe cryptography
- What (Solution)
    - Ability to store and manage QSA keys in the PKDS (asymmetric key store for ICSF).
- Wow (Benefit / Value, Need Addressed)
    - By storing the key material in the PKDS, many capabilities can be utilized:
        - Key store policy
        - Key lifecycle auditing
        - Key usage auditing
        - Better access control
        - Ability to transparently reencipher keys under a new master key

Usage & Invocation

- Several ICSF CCA APIs are updated to allow use of the PKDS label of a QSA token instead of a QSA token directly, just as already exists for RSA and ECC.
- For example, one can generate a Dilithium key, send the public key to a partner, sign a message, and then send the signature to a partner to be verified.
  - Partner 1
    - CSNDPKG (PKA Key Generate) to generate a QSA token
    - CSNDKRC to store the QSA token under a label
    - CSNDPKX to extract the public key to send to a partner
    - CSNDDSG to sign a message to send the message and signature to a partner
  - Partner 2
    - CSNDKRC to store the QSA public key under a label
    - CSNDDSV to verify the message and signature
- A full list of updated interfaces is on the next page

Usage & Invocation

- Using digital signatures
  - Digital Signature Generate (CSNDDSG and CSNFDSG)
  - Digital Signature Verify (CSNDDSV and CSNFDSV)
- Managing PKA cryptographic keys
  - PKA Key Generate (CSNDPKG and CSNFPKG)
  - PKA Key Import (CSNDPKI and CSNFPKI)
  - PKA Public Key Extract (CSNDPKX and CSNFPKX)
- Key data set management
  - Coordinated KDS Administration (CSFCRC and CSFCRC6)
  - Key Data Set List (CSFKDSL and CSFKDSL6)
  - Key Data Set Metadata Read (CSFKDMR and CSFKDMR6)
  - Key Data Set Metadata Write (CSFKDMW and CSFKDMW6)
  - Key Data Set Record Retrieve (CSFRRT and CSFRRT6)
  - Key Data Set Update (CSFKDU and CSFKDU6)
  - PKDS Key Record Create (CSNDKRC and CSNFKRC)
  - PKDS Key Record Read and PKDS Key Record Read2 (CSNDKRR or CSNDKRR2 and CSNFKRR or CSNFKRR2)
  - PKDS Key Record Write (CSNDKRW and CSNFKRW)

Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None beyond the existing hardware requirements for QSA
- Exploiters
  - No announced exploiters yet

Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Sysplex must be at the new z/OS level: No
- No upgrade actions required
- Coexistence APAR OA60857 on:
  - HCR77D1
  - HCR77D0
  - HCR77C1
  - HCR77C0
  - Coexistence is only required if the PKDS is converted to the larger format to exploit the new functionality and this PKDS is to be shared with lower-level systems.
- Note: the CKDS has been enabled for the larger format as well in anticipation of future requirements. No services have been updated to support larger tokens in the CKDS, but the CKDS may be converted earlier. The same coexistence APAR will support the larger LRECL CKDS.

Installation & Configuration

- No changes to installation or configuration are required.
- If wishing to exploit the new function, the PKDS must be converted to the larger format
  - sample CSFPKDS is shipped to allocate the larger LRECL PKDS
  - perform a Coordinated PKDS Conversion (ICSF panel option 2.2.6)
  - Coexistence APAR must be applied to systems wishing to share the larger LRECL PKDS
- As mentioned previously, the CKDS can be converted to the larger format at any time. The same basic process as used for the PKDS applies
  - sample CSFCKDS is shipped to allocate the larger LRECL CKDS
  - perform a Coordinated CKDS Conversion (ICSF panel option 2.1.6)
  - Coexistence APAR must be applied to systems wishing to share the larger LRECL CKDS

Summary

- A larger LRECL PKDS can be utilized to store QSA key material.
  - Existing callable services that currently support QSA key tokens will be enhanced to support PKDS labels as well.
  - Existing callable services, utilities, and dialogs that handle either the entire PKDS or labels within the PKDS will be enhanced to support PKDS labels for QSA key tokens.
- A larger LRECL CKDS is defined. While there is no exploitation defined yet, this is done to prepare for potential future enhancements.
  - Existing utilities and dialogs that handle the CKDS will be enhanced to support the larger LRECL.
- Coexistence APAR OA60857 will allow all in-support releases of ICSF to use the larger LRECL KDSs but not manage them.
  - Older releases can still perform all the same callable service processing with key material that was previously supported. Only KDS management or management of PKDS labels to QSA keys is disallowed.
  - This is the normal support mechanism, where the release that introduces functionality is the only release that can manage the new functionality, but older releases can continue to do what they have been doing.

Appendix

- Publications updated
  - Cryptographic Services Integrated Cryptographic Service Facility Overview
  - Cryptographic Services Integrated Cryptographic Service Facility Administrator's Guide
  - Cryptographic Services Integrated Cryptographic Service Facility System Programmer's Guide
  - Cryptographic Services Integrated Cryptographic Service Facility Application Programmer's Guide
  - Cryptographic Services Integrated Cryptographic Service Facility Messages