NYRUG 2018

Protecting the Keys to the (Crypto) Kingdom - Securing ICSF with RACF

Laura Sperling and John Reale, III
z/OS Defect Support – ICSF & RACF

lmdaniel@us.ibm.com
jreale@us.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

AIX*	Domino*	Language Environment*	SYSREXX	z10
BladeCenter*	DS6000	MVS	System Storage	z10 BC
BookManager*	DS8000*	Parallel Sysplex*	System x*	z10 EC
CICS*	FICON*	ProductPac*	System z	zEnterprise*
DataPower*	IBM*	RACF*	System z9	zSeries*
DB2*	IBM eServer	Redbooks*	System z10	
DFSMS	IBM logo*	REXX	System z10 Business Class	
DFSMSdss	IMS	RMF	Tivoli*	
DFSMShsm	InfiniBand	ServerPac*	WebSphere*	
DFSMSrmm				
DFSORT				

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Windows Server and the Windows logo are trademarks of the Microsoft group of countries.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

* Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g. zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Agenda

- What is ICSF?
- Terminology
- Key Areas and Functionality provided
- Key exploiters
- RACF Considerations
- References

What is Cryptography?

- **A set of techniques to scramble or disguise data**
  - Keep data confidential
  - Verify data integrity
- **Cryptographic functions fall into 4 categories:**
  - Encryption
  - Decryption
  - Hashing
  - Generating and verifying digital signatures
- **ICSF (Integrated Cryptographic Service Facility) provides applications with APIs to invoke cryptographic functions**
  - RACF restricts access to these services

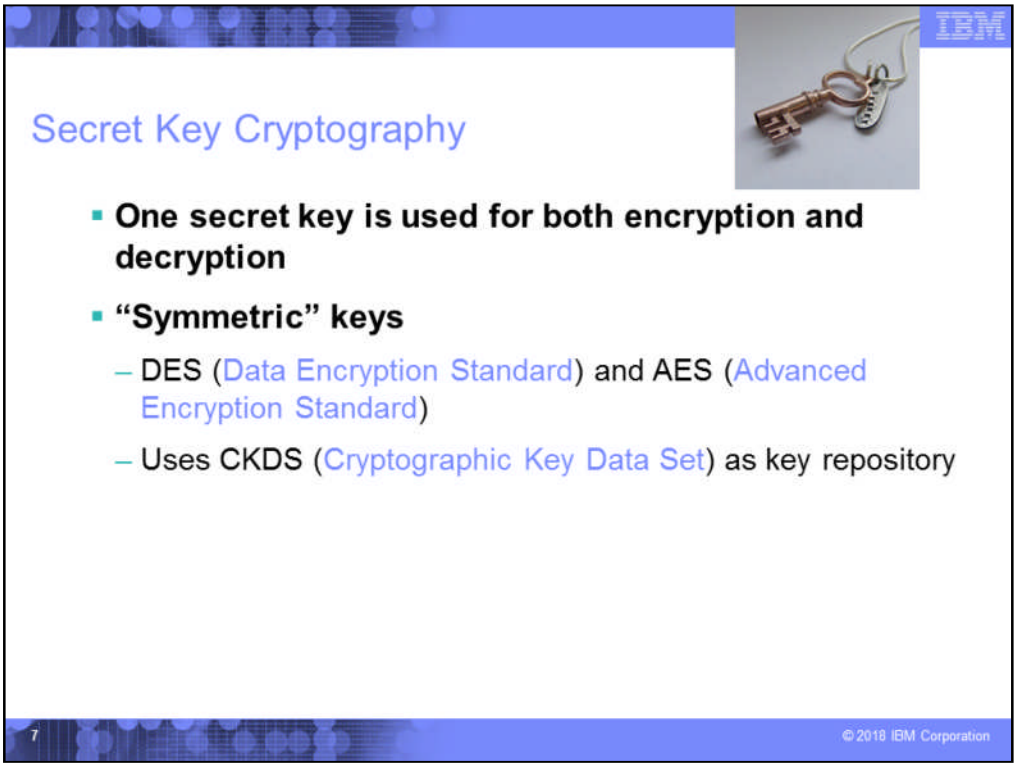
4

A valid digital signature gives a recipient reason to believe that the message was created by a known sender, such that the sender cannot deny having sent the message (authentication and non-repudiation) and that the message was not altered in transit (integrity). Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.

Why should a business run ICSF?

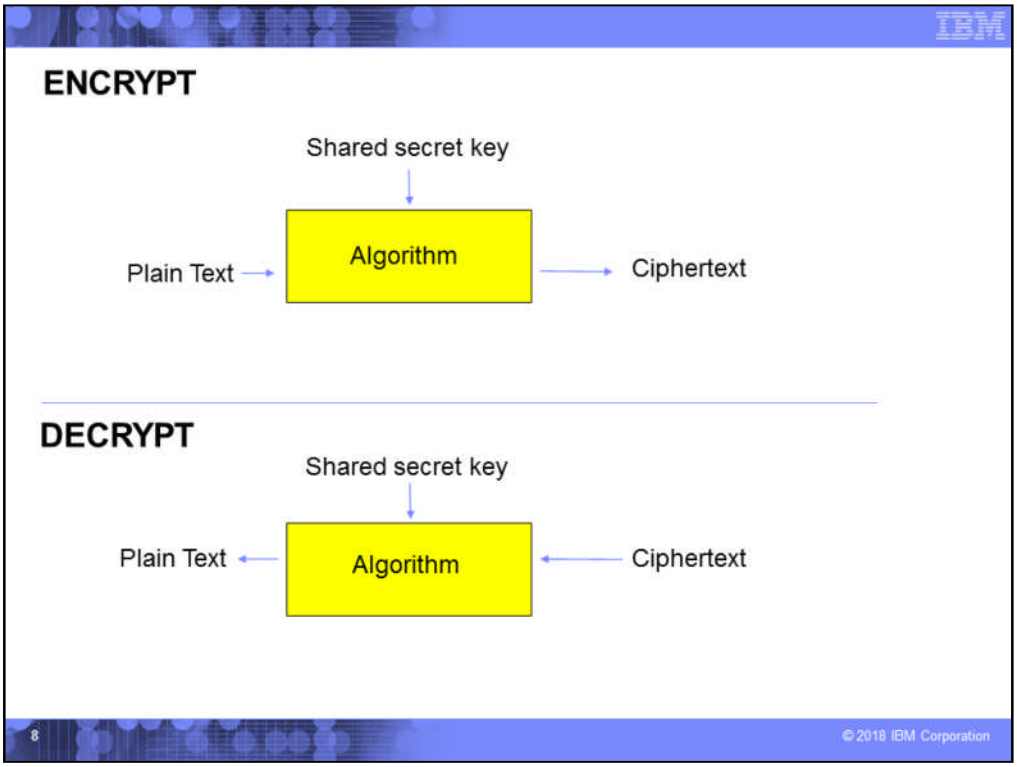
- **Data confidentiality and identity authentication**
  - Keep your data private
  - Transport data securely across a network
  - Exchange keys securely across a network
  - Verify data integrity and authenticity
  - Secure online credit card transactions
  - Use PINs for personal authentication
  - And much, much more!

Terminology



In a secret key system, it is critically important to maintain the secrecy of the shared key.

CKDS = Cryptographic Key Data Set, used to store both DES and AES keys, described in more detail later.



Secret key cryptography uses a conventional algorithm such as the Data Encryption Standard (DES) algorithm or the Advanced Encryption Standard (AES) algorithm that are supported by ICSF. Another term for secret key cryptography is symmetric cryptography. To have intelligent cryptographic communications between two parties who are using a conventional algorithm, this criteria must be satisfied: Both parties must use the same cryptographic algorithm.

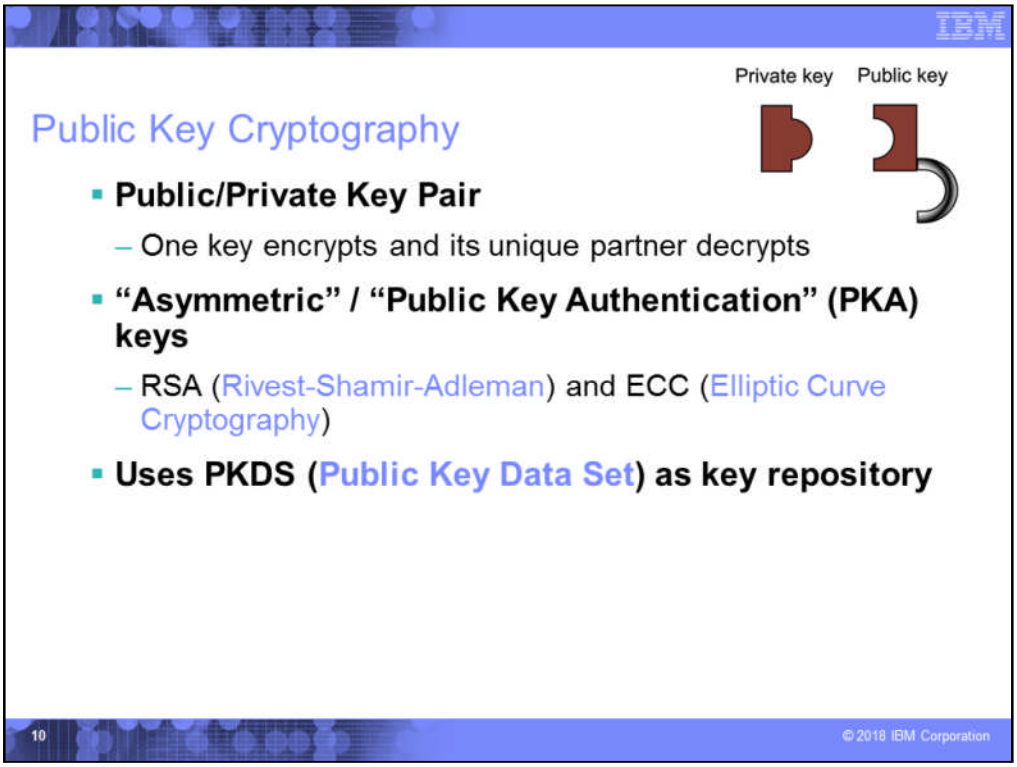
The cryptographic key that the sending party uses to encipher the data must be available to the receiving party to decipher the data.

https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb500/csfb500_Secret_key_cryptography.htm

Secret Key Cryptography (continued)

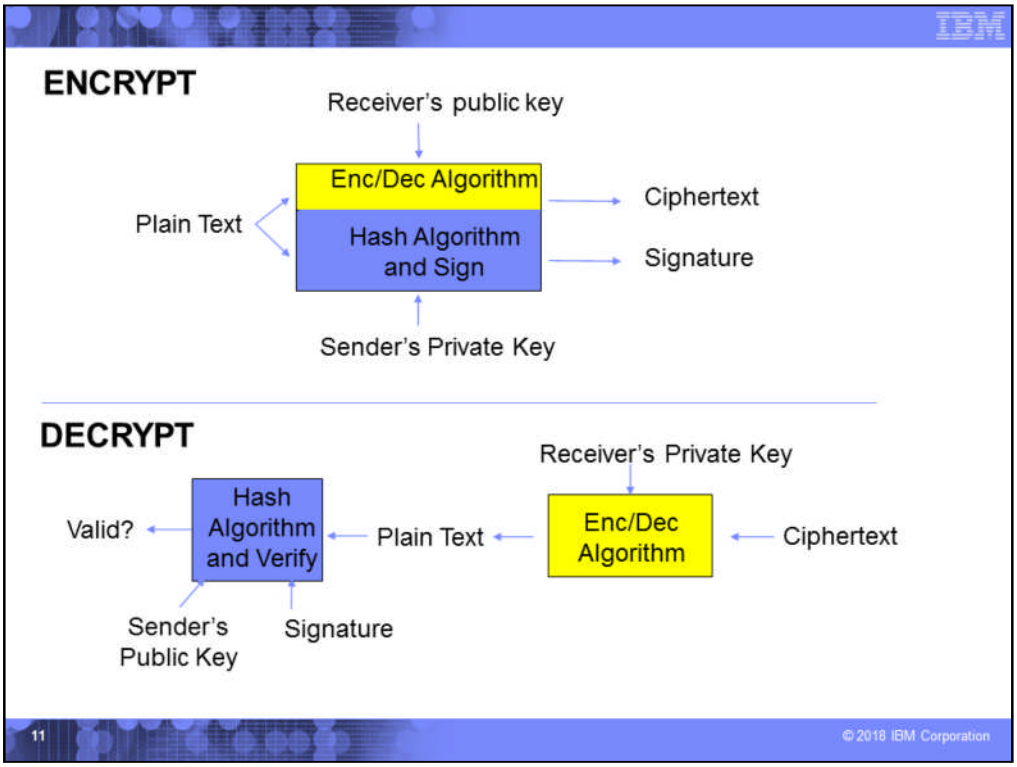
- **Benefit: Performance**
- **Common use: Flowing of secure data once SSL has created a secure connection (via handshake)**
- **Good to Know: DES -> AES**

DES is an inherently less secure algorithm than AES, causing many in the industry to move away from DES towards AES.



ICSF release HCR7770 was the last to require a DES Master Key be set.

Both public and private keys can be stored in PKDS.



Each party in a public key cryptography system has a pair of keys. One key is public and is published, and the other key is private.

The sending party looks up the receiving party's public key and uses it to encipher the data. The sender uses his or her private key to generate the associated digital signature.

The receiving party then uses its private key to decipher the data. The receiver also uses the sender's public key to verify the identity of the sender.

https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb500/csfb500_Public_key_cryptography.htm

Public Key Cryptography (continued)

- **Benefit: Validation, authenticity, privacy**
- **Common use: SSL handshake**
- **Good to Know:**
  - Numerous algorithms (not limited to key types)
  - Private keys MUST be kept VERY PRIVATE

Ideally, no human eyes should ever see clear contents of a private key! Create and store encrypted on z/OS within ICSF.

Public key cryptography requires complex mathematical calculations and is therefore minimally used in performance paths.

Key Terminology

- **Master Key**
  - Top-level key
  - Protects all keys of same type in the Key Data Set
    - 4 Types: DES, AES, RSA, ECC
  - Never in the clear!
    - Resides securely in tamper-proof cryptographic coprocessors
- **Operational Key**
  - Anything else!

Key Terminology (continued)

- **Secure Key**
  - Any key encrypted under the Master Key. Never appears 'in the clear' outside of a crypto coprocessor
- **Clear Key**
  - Any key not protected by encryption under another key
- **Protected Key**
  - A symmetric key encrypted under a CPACF-specific wrapping key and stored by an application
  - RACF determines whether a key can be used for wrapping and also which keys can be wrapped

A protected key can be a DES, TDES or AES key. Once a key has been wrapped, it can make use of CPACF directly, using the hardware to directly access a subset of symmetric key operations.

Key Terminology (continued)

- **Key Token**
  - A data structure that contains information about a key and usually contains a key or keys
  - Can be stored in KDS under a label
  - Can be saved elsewhere by application
  - Not same as PKCS#11 token

Types of Key Data Sets (ICSF)

- **Cryptographic Key Data Set (CKDS)**
  - Contains symmetric cryptographic keys (DES/AES)
- **Public / PKA Key Data Set (PKDS)**
  - Contains asymmetric cryptographic keys (RSA/ECC)
- **Token Key Data Set (TKDS)**
  - Contains PKCS #11 tokens / objects
  - Keys stored here may not be encrypted; RACF protection needed
- **An entry in a KDS also contains information about the key**

16

All Key Data Sets (KDS) are VSAM data sets.

ICSF provides a KGUP utility to allow for loading of clear keys into a Key Data Set.

Cryptographic Coprocessor ("The Card")

- **A microprocessor that adds cryptographic processing functions**
- **Master keys are stored here**
- **Tamper-resistant chip built into processor board**
  - Zeroizes saved Master Key data if removed improperly or if tampering is detected
- **Modern Types**
  - CEX5C, CEX6C

17

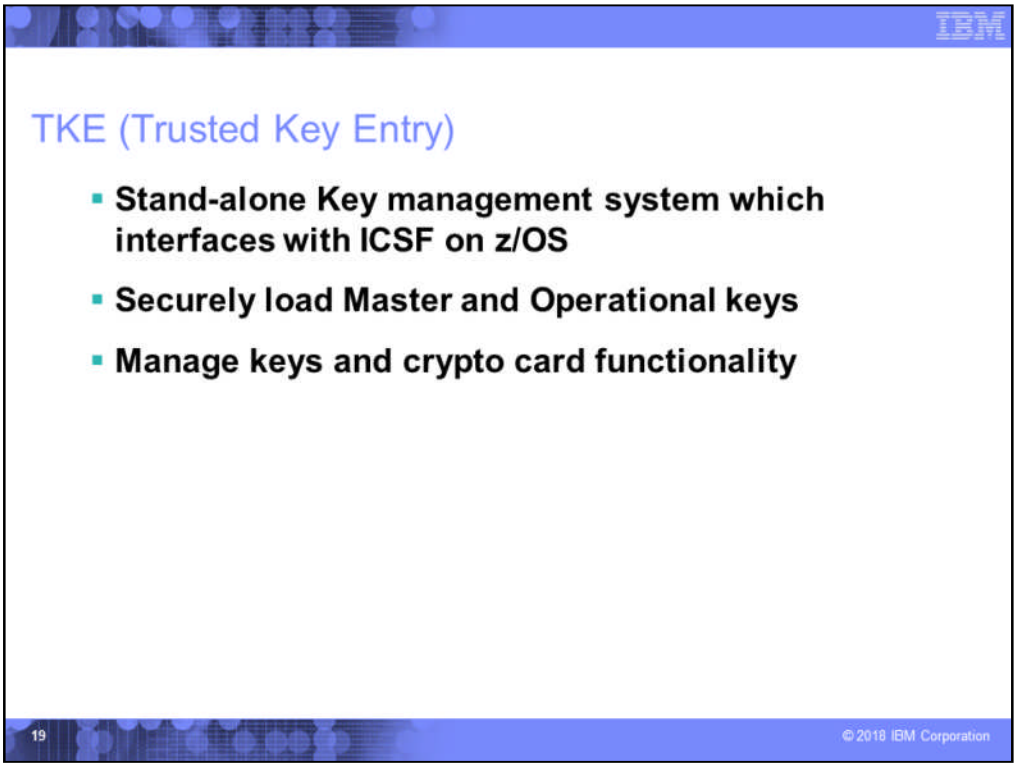
A cryptographic coprocessor allows keys stored within the KDS to be encrypted under respective Master Key at all times, thus never being exposed in the clear. We call these "Secure Keys".

A cryptographic coprocessor is also known as a "card" or "crypto card"

CP Assist for Cryptographic Functions (CPACF)

- **A set of cryptographic instructions available on General Purpose CPs**
- **Provides symmetric crypto functionality**
- **Not a Cryptographic Coprocessor**
  - Cannot be used with secure keys
  - Any application can call these instructions

The presence of CPACF with the SSL SPE APAR OA54127 applied will allow SSL to make use of stronger symmetric algorithms, even when SSL Security Level 3 FMID not installed.



Additional information about TKEs can be found in the ICSF TKE Workstation User's Guide:
https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb600/toc.htm

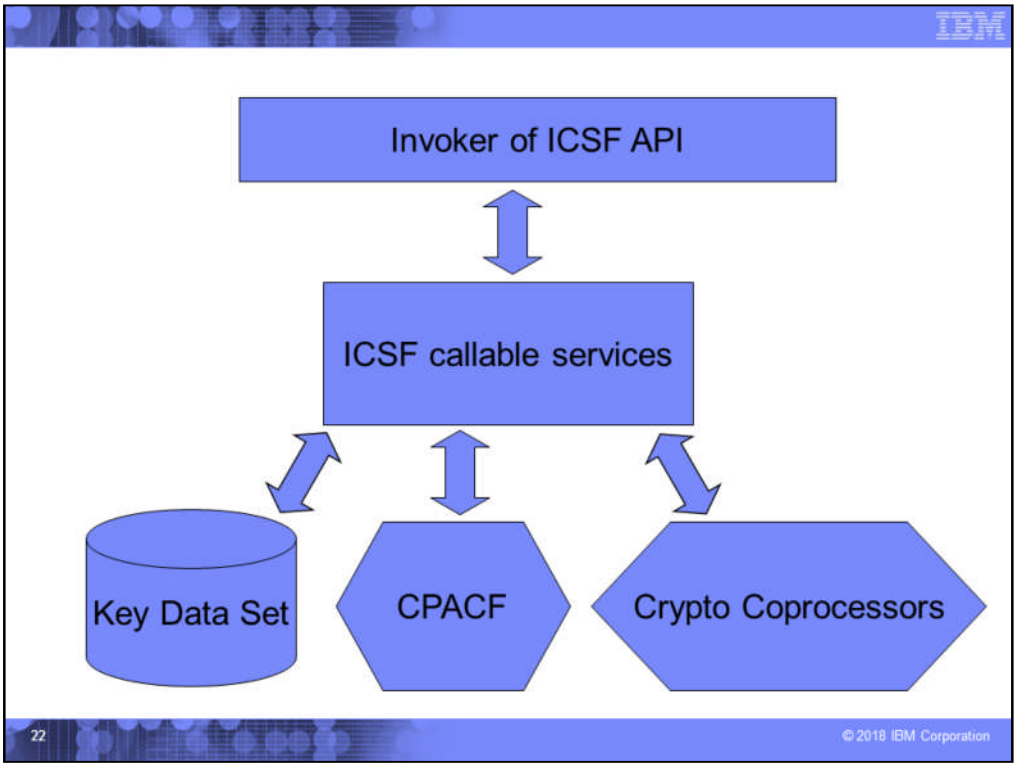
Integration into z/OS

How can ICSF services be invoked?

- **Callable Services**
- **Panel-driven Services**
- **Remotely through TKE**
- **Via JCL (i.e. KGUP)**

- **RACF manages access to all methods!**

- **Full functionality requires the use of crypto cards**
- **Clear key operation functionality without cards**



Here is a pictorial representation of how an application would call an ICSF API, and how ICSF satisfies the request. Depending upon the API call being processed, ICSF may need access to the KDS and/or crypto coprocessors.

Key Exploiters

- **Pervasive Encryption**
- **System SSL**
- **IMS**
- **DB2**
- **z/OS UNIX**
  - Random number generation (CSNBRNG) via use of /dev/random and /dev/urandom
- **Customer applications**

Each application exploiting ICSF may have unique access requirements. Refer to each application's documentation to learn more.

RACF Considerations

Strategies

- **Performance and Trust**
- **How much protection:**
  - Keys?
  - Services?
  - Labels?
  - Tokens?

More information can be found in the z/OS ICSF Administrator's Guide, section "**Steps for SAF-protecting ICSF services and CCA keys"**

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.csfb300/csfb300_Steps_for_RACF-protecting_keys_and_services.htm

ICSF Config Option: CHECKAUTH

- **Should ICSF perform access control checking of Supervisor State and System Key callers?**
- **Specified in ICSF Options Data Set**
- **YES**
  - RACROUTE requests issued
    - Enforce profile checking for *ALL* callers
  - May be logged in RACF SMF Records
- **NO (default)**
  - Boosts performance
  - RACROUTE *not* issued
    - CSFSERV and CSFKEYS profiles not examined
  - No SMF records

CHECKAUTH(YES) applies to both CSFSERV class and CSFKEYS class checks. When YES is specified, LOG=ASIS is used.

RACF Classes used for ICSF

- **CSFSERV**
  - ICSF callable services
- **CRYPTOZ**
  - PKCS#11 tokens
- **CSFKEYS**
  - Key labels
- **XCSFKEY**
  - Export of key labels
- **XFACILIT**
  - Key Store Policy controls

Key Store Policy (KSP)

- Set of controls (in XFACILIT class) that manages use of crypto keys in various forms and situations
- Some controls 'activate' KSP
- Some 'require' KSP to be activated
- Some do neither!

z/OS 2.3 ICSF Administrator's Guide Key Store Policy Overview:

https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb300/defksp.htm

Key Store Policy

- **Main purposes**
  - CKDS/PKDS Tokens (A)
  - CKDS/PKDS Duplicate Tokens (A)
  - CKDS/PKDS Default Tokens (R)
  - Granular Key Authority levels
  - Symmetric Key Label Export Control via XCSFKEY
  - PKA Key Mgmt Extensions (R)
  - Using Archived Keys

A = Activates KSP when specified
R = Requires KSP to be active prior to use

Enabling any one of the following controls will activate Key Store Policy for a CKDS:

CSF.CKDS.TOKEN.CHECK.LABEL.WARN

CSF.CKDS.TOKEN.CHECK.LABEL.FAIL

CSF.CKDS.TOKEN.NODUPLICATES

Similar profiles for a PKDS.

CSFSERV Resource Class

- **Restricts access to ICSF Services**
- **If no profile, access is granted**
- **Example**
  - RDEFINE CSFSERV CSFENC UACC(NONE)
  - RDEFINE CSFSERV CSFKGUP UACC(NONE)
  - RDEFINE CSFSERV CSFSMK UACC(NONE)

30

More information can be found in the z/OS ICSF Administrator's Guide, section "**Setting up profiles in the CSFSERV general resource class"** https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.csfb300/ctlserv.htm

Users can be permitted or denied access to the KGUP utility via the CSFKGUP profile in the CSFSERV class.

More information can be found in the z/OS Writing PKCS#11 Applications manual, Chapter 1 “Overview”:

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.csfba00/token_overview.htm and definitions of

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.csfba00/control_access.htm

Reminder, PKCS#11 tokens live in the TKDS.

RACF calls in the CRYPTOZ class use LOG=NOFAIL.

CRYPTOZ Resource Class (Continued)

- **A resource for each role**
  - SO.*token-name*
    - Controls access of the Security Officer (SO) role to the token
    - READ – Weak SO
    - UPDATE – SO R/W
    - CONTROL – Strong SO
  - USER.*token-name*
    - Controls access of the User role to the token
    - READ – User R/O
    - UPDATE – Weak User
    - CONTROL – User R/W

These roles are defined here:

https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb300/racfprot.htm

CSFKEYS Resource Class

- **Restricts access to ICSF Key Labels stored in CKDS and PKDS**
- **Profile Name:** Key Label Name
- **By default:**
  - READ: Read, Write, Create and Delete a label
- **Example:**
  - RDEFINE CSFKEYS JOHNS.LABEL UACC(READ)
- **Might want more granular access requirements…**

33

More information can be found in the z/OS ICSF Administrator's Guide, section "**Setting up profiles in the CSFKEYS general resource class"**

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.csfb300/ctlkey.htm

CSFKEYS – Granular Key Label Access Checking

- **Stand-alone KSP feature**
- **Optional KSP profiles in the XFACILIT class**
  - CSF.CSFKEYS.AUTHORITY.LEVELS.WARN
  - CSF.CSFKEYS.AUTHORITY.LEVELS.FAIL
- **Granular access requirements**
  - READ: Read from label
  - UPDATE: Create or write to label
  - CONTROL: Delete a label
- **Example**
  - PERMIT CLASS(CSFKEYS) JOHNS.LABEL ACC(UPDATE) ID(REALE)

34

Granular Key Label Access checking only works when an ICSF service is passed a key label.

Key Store Policy Information:
https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb300/defksp.htm

CSFKEYS – via Tokens

- **XFACILIT profiles**
  - CSF.CKDS.TOKEN.CHECK.LABEL.WARN / CSF.CKDS.TOKEN.CHECK.LABEL.FAIL
  - CSF.PKDS.TOKEN.CHECK.LABEL.WARN / CSF.PKDS.TOKEN.CHECK.LABEL.FAIL
- **Initiates translation from Token to Label for the RACF Auth check**
- **Activates KSP when specified**

35

For a more robust solution, Key Store Policy should also be enabled for both CKDS and PKDS to allow the increased authority checking when tokens are passed in, as well. Key Store Policy can be turned on using:

CSF.CKDS.TOKEN.CHECK.LABEL.WARN /
CSF.CKDS.TOKEN.CHECK.LABEL.FAIL and
CSF.PKDS.TOKEN.CHECK.LABEL.WARN /
CSF.PKDS.TOKEN.CHECK.LABEL.FAIL

CSFKEYS – Default Tokens

- **Stand-alone KSP feature**
- **XFACILIT profiles**
  - CSF.CKDS.TOKEN.CHECK.DEFAULT.LABEL
  - CSF.PKDS.TOKEN.CHECK.DEFAULT.LABEL
- **Translates a Token without a matching Label to:**
  - CSF-CKDS-DEFAULT profile in CSFKEYS
- **Requires KSP to be active prior to use**

There may not be a label found for a given token. Perhaps the application is managing its tokens independently.

CSFKEYS – Duplicate Keys

- **XFACILIT profiles**
  - CSF.CKDS.TOKEN.NODUPLICATES
  - CSF.PKDS.TOKEN.NODUPLICATES
- **Will not allow one Token to be associated with multiple Labels / keys**
- **Activates KSP when specified**
- **The CSFDUTIL utility reports on existing dupes.**

This will prevent a single token from being associated with multiple key labels.

XCSFKEY - Symmetric Key Label Export Control

- **Stand-alone KSP feature**
- **KSP XFACILIT profiles to activate**
  - CSF.XCSFKEY.ENABLE.AES
  - CSF.XCSFKEY.ENABLE.DES
- **Increases access required to a symmetric key label for export purposes**
- **Profiles in XCSFKEY use same format as CSFKEYS (shown earlier)**

38

ICSF Admin Guide: "**Increasing the level of authority needed to export symmetric keys**":

https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb300/indksp.htm

CSFKEYS – PKA Key Mgmt Extensions

- **XFACILIT Profile (to enable)**
  - CSF.PKAEXTNS.ENABLE
- **Place additional restrictions on how keys can be used**
- **Each extension involves an option in the ICSF segment of the CSFKEYS / XCSFKEY profiles**
- **Requires KSP to be active prior to use**

A CSFKEYS profile can contain an ICSF segment, which specifies rules for key use. Setting restrictions can help ensure that keys are used only for intended purposes, regardless of who has access to the keys.

PKA Key Mgmt Extensions - options

- **Restrict Asymmetric from Import/Export Ops**
- **Restrict Asymmetric from Handshake Ops**
  - Both involve the ASYMUSAGE field
- **Example**

```
RALTER CSFKEYS JOHNS.LABEL
    ICSF(ASYMUSAGE(NOSECUREEXPORT))
RALTER CSFKEYS JOHNS.LABEL
    ICSF(ASYMUSAGE(NOHANDSHAKE))
SETROPTS RACLIST (CSFKEYS) REFRESH
```


The ASYMUSAGE field enables you to restrict asymmetric keys covered by the profile from being used in:

1. Secure import and export operations
2. Handshake operations

PKA Key Mgmt Extensions - options

- **Restrict Symmetric from Exports**
- **and by which Asymmetrics**
  - The SYMEXPORTABLE field
  - With options BYNONE, BYLIST, BYANY
- **Example**

```
RALTER CSFKEYS JOHNS.LABEL
    ICSF(SYMEXPORTABLE(BYNONE))
SETROPTS RACLIST (CSFKEYS) REFRESH
```

Archived Keys

- **Stand-alone KSP feature**
- **XFACILIT Profile**
  - CSF.KDS.KEY.ARCHIVE.USE
- **Will not fail a request that uses a Key that has been archived**

Before we had KSP…

- **Two options for ReWrapping:**
  - SYMCPACFWRAP: can an encrypted key be rewrapped
  - SYMCPACFRET: can a protected key be returned
- **Example**

```
RALTER CSFKEYS JOHNS.LABEL ICSF(SYMCPACFWRAP(YES))
SETROPTS RACLIST (CSFKEYS) REFRESH
```


The SYMCPACFWRAP field of the ICSF segment enables the covered encrypted key to be rewrapped (protected) using the CPACF wrapping key. The specification:

SYMCPACFWRAP(YES) indicates that encrypted keys covered by the profile can be rewrapped.

SYMCPACFWRAP(NO), which is the default, indicates that encrypted keys covered by the profile cannot be rewrapped.

* If your installation requires that a particular encrypted key must never exist outside of the tamper-resistant hardware boundary, do not use the SYMCPACFWRAP(YES) specification in the CSFKEYS profile that covers the key.

The SYMCPACFRET field is needed for CSNBKRR2 (Key Record Read 2) to allow a protected key to be returned to the application.

https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb300/enuenc.htm

Miscellaneous

- **CKDS browser added in HCR77C1 (CSFBRCK) uses LOG=NONE on CSFSERV**
  - Service allows browsing of symmetric keys stored in CKDS

More information about setup and usage can be found in the ICSF Administrator's Guide.

References

- **ICSF Overview:**
  - http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/CSFB5ZA1
- **ICSF TKE Workstation User's Guide:**
  - http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/CSFB6Z50
- **A Clear Key/Secure Key/Protected Key Primer:**
  - https://www-03.ibm.com/support/techdocs/atsmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/011df1542eb4b83b86257eca005ad4c7/$FILE/Clear%20Key%20Secure%20Key%20Protected%20Key%20Primer_100720.pdf

Appendix – How to get started?

- **Review ICSF Administrator's Guide, section "SAF Controls":**
  - https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.csfb300/ctl.htm
- **Key Store Policy link**
  - https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb300/defksp.htm