# Intro to z/OS Crypto and ICSF

*Ross Cooper, CISSP®*
*IBM Corporation*

*March 2nd, 2015*
*Session: 16777*

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides **education**, professional **networking** and industry **influence**.

# Topics:

- Intro to Cryptography:
  - Asymmetric & Symmetric Cryptography
  - Hashing, Digital Signatures
  - Digital Certificates
- Overview of ICSF
  - Hardware cards
  - CPACF
  - ICSF Key Types
  - TKE
- Other cryptography functions available on z/OS:
  - Java Crypto Providers, System SSL, PKI Services...

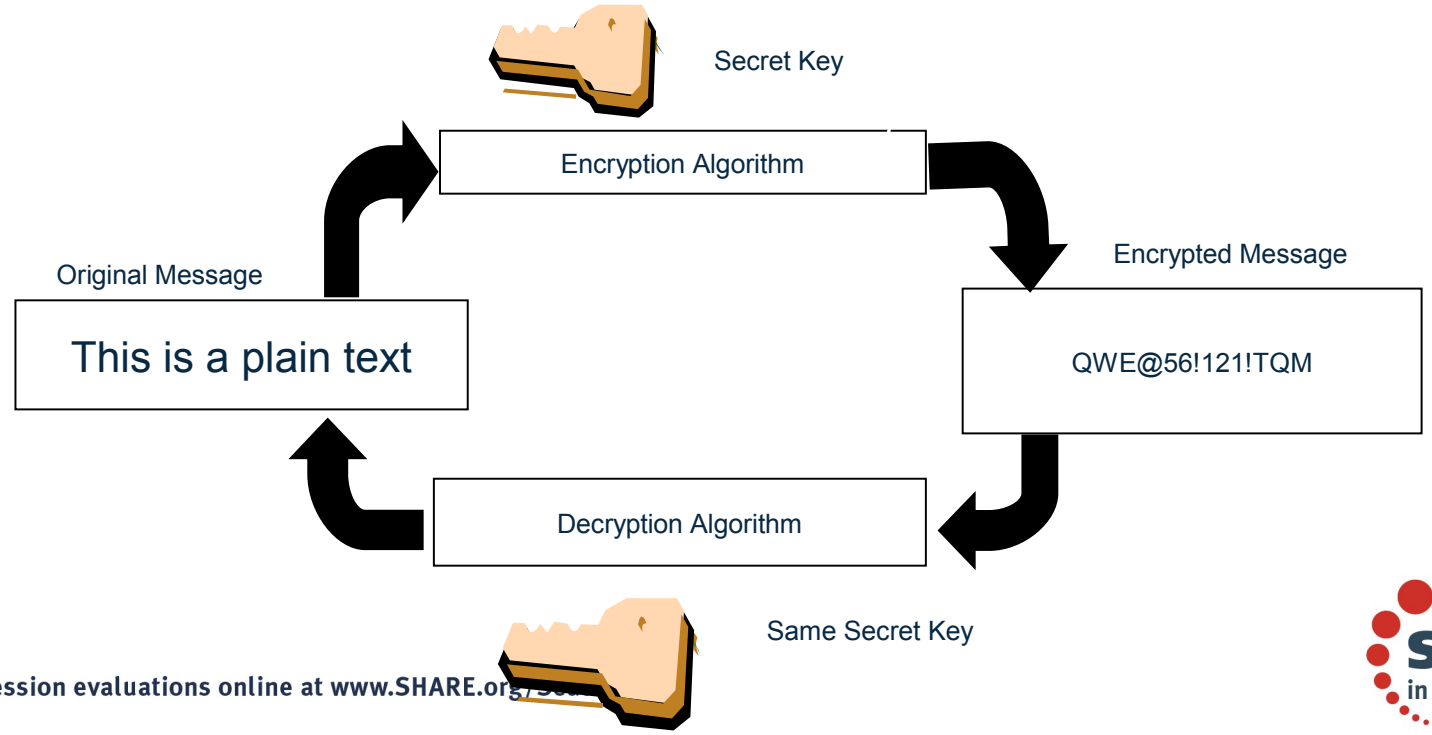
# Intro to Crypto

# Cryptography

- Cryptography is a set of techniques used to provide the following services:
  - **Data Confidentiality**
    - Protecting data from disclosure to unauthorized parties
      - Symmetric Encryption
      - Asymmetric Encryption
  - **Data Integrity**
    - Modification Detection,
      Message Authentication,
      Non-repudiation
      - Digital Signatures (Hashing + Asymmetric Encryption)

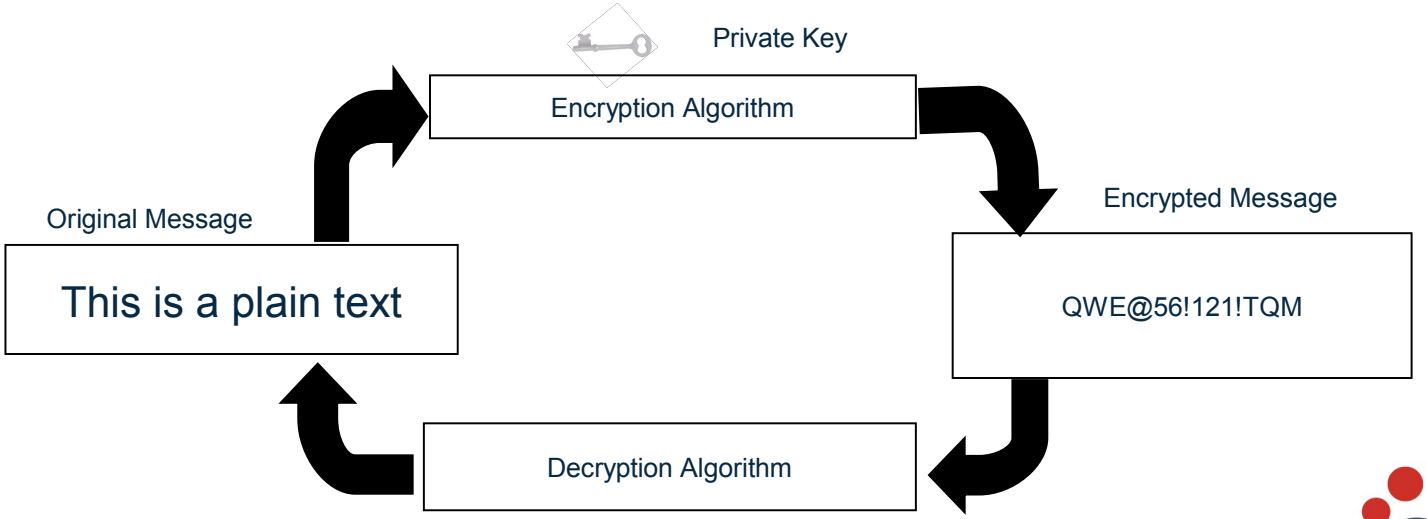
# Symmetric Encryption

- **Provide data confidentiality**
- **Same key** used for both encryption and decryption
- **Fast**, used for bulk encryption/decryption
- **Securely sharing** and exchanging the key between both parties is a major issue
- **Common algorithms**: DES, Triple DES, AES



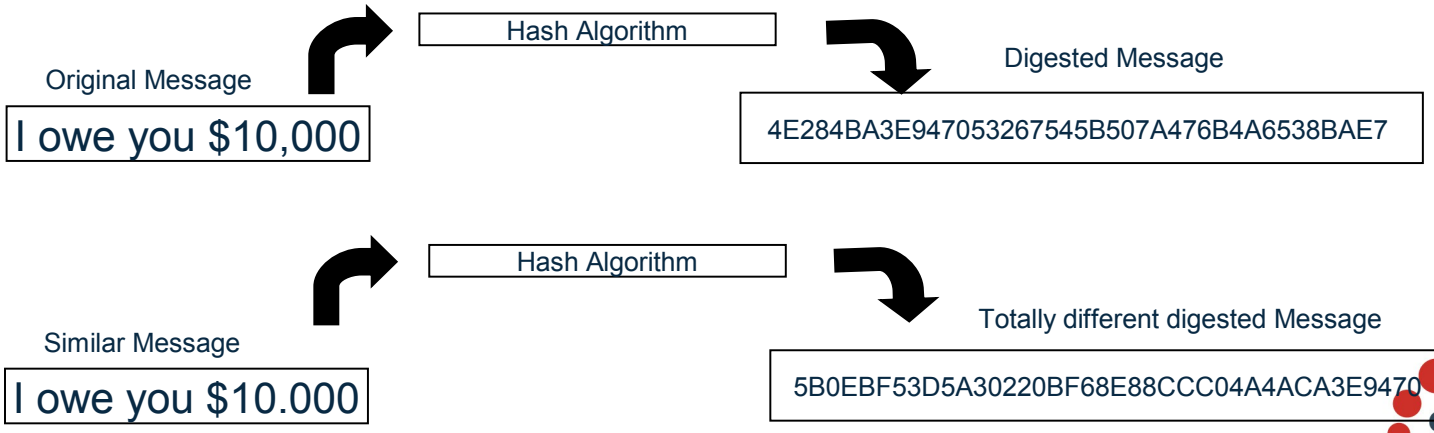
# Asymmetric Encryption

- **Public / private key pairs** - 2 different keys
- A public key and a related private key are **numerically associated** with each other.
- Provide data **confidentiality**, **integrity** and **non repudiation**
- **Data encrypted/signed using one** of the keys may only be **decrypted/verified using the other** key.
- **Slow,** Very expensive computationally
- **Public key is freely distributed** to others, private key is securely kept by the owner
- **Common algorithms**: RSA, DSA, ECC



# Message Digest (Hash or Fingerprint)

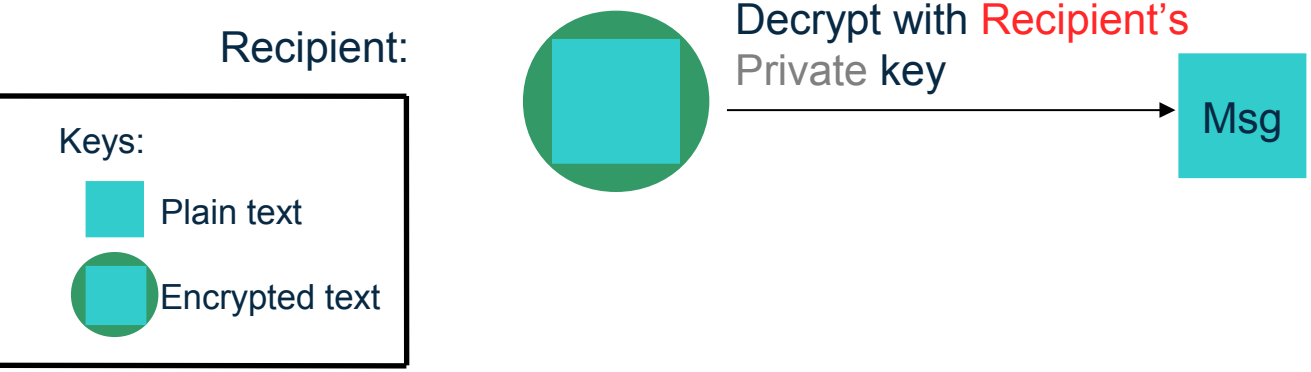
- A **fixed-length value** generated from **variable-length data**
- Unique:
  - The same input data always generates the same digest value
  - Tiny change in data causes wide variation in digest value
  - Theoretically impossible to find two different data values that result in the same digest value
- **One-way**: can't reverse a digest value back into the original data
- **No keys involved** – Result determined only by the algorithm
- Play a part in data integrity and origin authentication
- **Common algorithms**: SHA1, SHA256



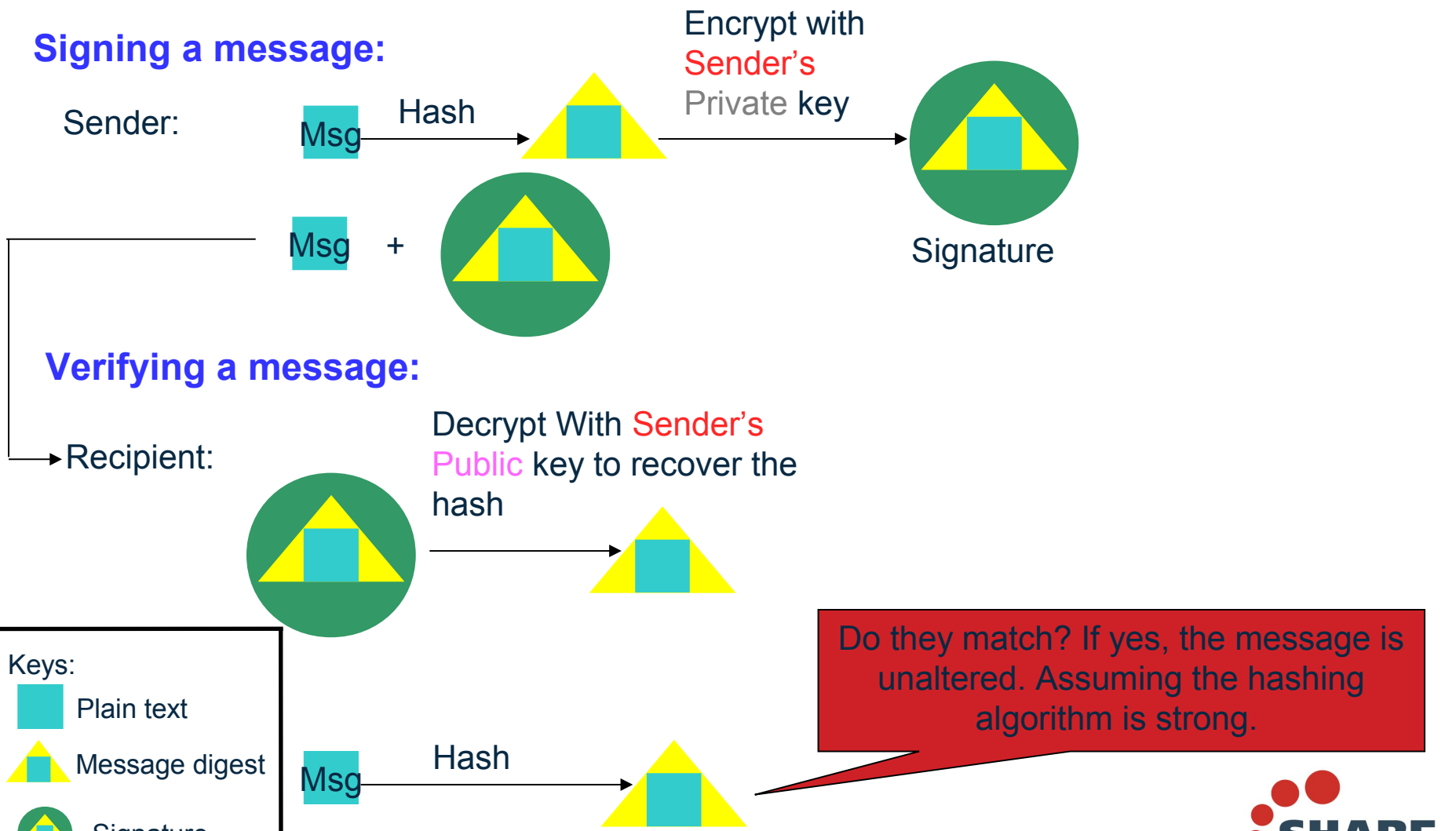
# Asymmetric Encryption (for confidentiality)

## Encrypting a message:


## Decrypting a message:



# Signing (for integrity and non repudiation)



# Digital Certificates

- A Digital Certificate is a digital document issued by a trusted third party which binds an end entity to a public key.

- **Digital document:**
  - Contents are organized according to ASN1 rules for X.509 certificates
  - Encoded in binary or base64 format
- **Trusted third party aka Certificate Authority (CA):**
  - The consumer of the digital certificate trusts that the CA has validated that the end entity is who they say they are before issuing and signing the certificate.
- **Binds the end entity to a public key:**
  - **End entity** - Any person or device that needs an electronic identity. Encoded in the certificate as the Subjects Distinguished Name (SDN). Can prove possession of the corresponding private key.
  - **Public key** - The shared half of the public / private key pair for asymmetric cryptography
  - **Digitally signed by the CA**

# Digital Certificates - Usage

- **Prove Identity to a peer:**
  - Owner of the certificate can prove possession of the certificate's private key
  - Identity can be validated by checking it is signed by a trusted Certificate Authority
- **Prove authenticity of a digital document:**
  - Programs can be signed by code signing certificates
  - E-mail signatures
  - Certificates are signed by CA certificates
- **Establish a secure connection:**
  - Certificates contain a public key which allows protocols such as SSL and TLS to exchange session keys

# Intro to ICSF

# ICSF

- Integrated Cryptographic Service Facility (ICSF)
  - Base element of z/OS that provides cryptographic services
- Provides an application programmers interface (API) for applications that need to perform crypto
- Provides basic key management
- Keystores (CKDS, PKDS, TKDS) for cryptographic key material
- Provides access:
  - Hardware Cryptographic Coprocessors, Cryptographic Accelerators
  - CP Assist for Cryptographic Function (CPACF)

# ICSF Services

- **Standard Cryptographic Functions:**
  - Encryption and Decryption of Data
  - Hashing algorithms
  - Digital signatures
  - Message Authentication Codes (MACs)
  - Key generation and distribution
- **Financial Institution Services:**
  - Personal Identification Numbers (PINs)
  - Card-verification values
  - Translation of data and PINs in networks
  - ATM remote key loading
  - EMV functions for integrated circuit card specifications
  - Secure Electronic Transaction
- **Protocols and Standards:**
  - Secure Sockets Layer
  - PKCS #11


# Cryptographic Coprocessors

- ICSF uses **PCIe Cryptographic Coprocessors** to perform hardware crypto functions
- These cards provide a high-security, high-throughput cryptographic subsystem.
- The hardware security modules are validated to FIPS 140-2, Overall Level 4 (highest level of security).
- Features:
  - Tamper responding, programmable, cryptographic PCIe cards, containing CPU, encryption hardware, RAM, persistent memory, a hardware random number generator, time of day clock and infrastructure firmware.

# Cryptographic Coprocessors

- Cards can be in three configuration modes:
  - **Coprocessor Mode:** IBM Common Cryptographic Architecture
  - **Accelerator Mode:** Supports RSA clear key and SSL acceleration
  - **Enterprise PKCS#11 (EP11) Mode:** PKCS#11 programming interface
- The firmware running in the coprocessor can be customized to meet special requirements
- Custom firmware loads are called User Defined Extensions (UDXs)
- Several algorithms are supported in hardware:
  - DES/TDES  MAC/CMAC
  - AES, AESKW, AES GMAC, AES GCM, AES XTS mode, CMAC
  - MD5, SHA-1, SHA-2 (224,256,384,512), HMAC
  - RSA (512, 1024, 2048, 4096)
  - Montgomery Modular Math Engine
  - RNG (Random Number Generator)
  - Clear Key Fast Path (Symmetric and Asymmetric)

# Cryptographic Accelerators

- High performance RSA asymmetric algorithms
- Designed for **maximum Secure Socket Layer** (SSL) acceleration rather than for specialized financial applications and secure key processing
- Can support over 2000 SSL handshakes per second
- Previously shipped as a separate hardware feature
- Cryptographic Coprocessors can be configured to become Cryptographic Accelerators

# ICSF Master Keys

- **Stored within the secure hardware boundary of the cryptographic coprocessor**
- **ICSF uses five master keys to protect operational keys:**
  - **DES Master Key** (DES-MK aka SYM-MK) - 128 bit key (or now 256-bit)
    - Protects DES/TDES (symmetric) application keys
  - **AES Master Key** (AES-MK) - 256 bit key
    - Protects AES (symmetric) application keys
  - **Asymmetric-keys master key** (RSA-MK aka ASYM-MK) - 192 bit key
    - Protects RSA (asymmetric) private keys
  - **Elliptic Curve Master Key** (ECC-MK) - 256 bit key
    - Protects ECC (asymmetric) private keys
  - **Enterprise PKCS #11 Master Key** (P11-MK) - 256 bit key
    - Protects PKCS #11 keys

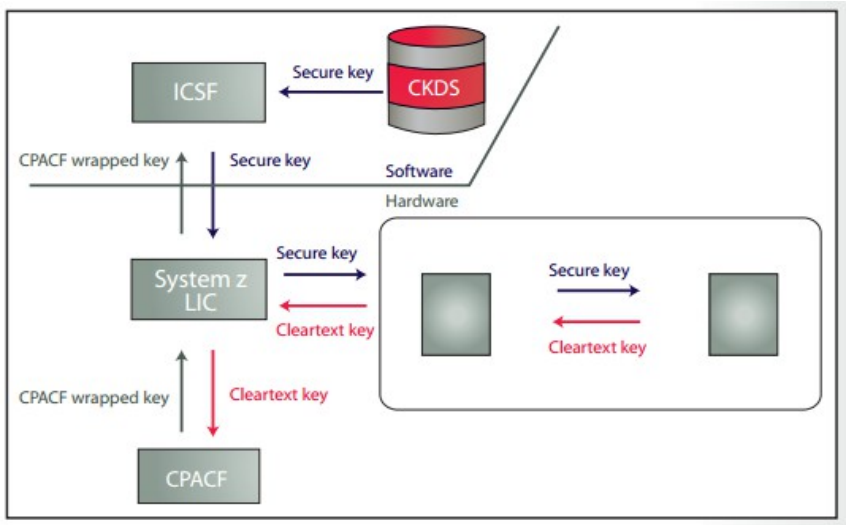
# CP Assist for Cryptographic Function (CPACF)

- Encryption accelerator functionality is provided on a quad-core chip, which is designed to provide the following high-speed cryptography functions:
  - **Data Encryption Standard** (DES) 56-bit key
  - **Triple Data Encryption Standard** (TDES) 168-bit keys
  - **Advance Encryption Standard** (AES) for 128-bit, 192 and 256 bit keys
  - **Secure Hash Algorithm** (SHA) SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512
  - **Pseudo Random Number Generation** (PRNG)
  - **Protected Key Support**

- Access provided via assembler instructions

# Clear Key, Secure Key and Protected Key

- **Clear Key** – when performing symmetric encryption, TDES and AES, with clear keys, ICSF uses the CPACF to provide high performance. Clear Key refers to key material that is in the clear, meaning the clear key value appears within application storage and within the keystore
- **Secure Key** - provides high security because the key material is protected by the master key. Master keys are loaded within the cryptographic coprocessor and are used to wrap and unwrap secure key material within the secure boundaries of the HSM. This prevents secure key material from ever appearing in the clear.
- **Protected Key** - provides a high performance and high security solution by taking advantage of the high speed CPACF while utilizing symmetric keys protected by the cryptographic coprocessor Master Key. To use a CKDS encrypted key, the ICSF segment of the CSFKEYS class general resource profile associated with the specified key label must contain SYMCPACFWRAP(YES).



# ICSF Web Deliverables

- As new cryptographic hardware becomes available, ICSF is updated and new functions are delivered via web deliverables outside of the z/OS release cycle
- Web deliverables along with their associated ICSF release publications are available on the z/OS downloads website:
  - **http://www.ibm.com/systems/z/os/zos/downloads**
- Each new release of z/OS contains a version of ICSF incorporated into its base, however this may not be the latest and greatest level of ICSF
- Check the z/OS downloads website for the latest level of ICSF
- It is recommended to run with the latest level of ICSF to ensure you have all of the latest and greatest features

# Using RACF to Protect Keys and Services

- **RACF** can be used to protect and audit the use of ICSF keys and services
- The **CSFKEYS** class controls access to cryptographic keys in the CKDS and PKDS
  - Create profiles based on CKDS and PKDS key labels
- The **CSFSERV** class controls access to ICSF services and ICSF TSO panel utilities
- The **XCSFKEY** class is used to control the transfer of secure AES and DES keys from encryption under the MK to encryption under an RSA key
  - This is used for authorization checking of the Symmetric Key Export service
- The **CRYPTOZ** class controls access to, and defines a policy for PKCS #11 tokens which are used by ICSF's PKCS #11 callable services.
- **Recommendation:**
  - Make sure that access is granted only to the processes and people who need access.

# CCA Access Control Points

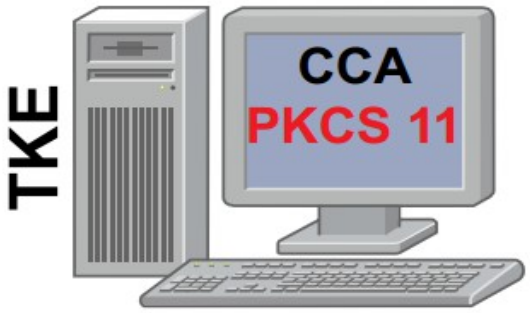
- Access to services that are executed on the CCA coprocessor is through Access Control Points in the ICSF Role.
- To execute services on the coprocessor, access control points must be enabled for each service in the ICSF Role.
- The TKE workstation allows you to enable or disable access control points.
- For systems that do not use the optional TKE Workstation, most access control points (current and new) are enabled in the ICSF Role with the appropriate licensed internal code on the coprocessor.
- See the table in the ICSF Admin Guide for a list of access control points and the default setting of each access control point.
- New TKE users and non-TKE users have the default set of access control points enabled.
- Existing TKE users who have changed the setting of any access control point, any new access control points will not be enabled.
- Recommendation:
  - Only enable ACPs for callable services you use.

# Key Store Policy

- Key Store Policy allows you to control how encrypted key tokens defined in the CKDS and PKDS can be accessed and used.
- Key Store Policy is defined using resource profiles in the **XFACILIT** class
- Key Store Policy controls allows you to:
  - Verify that a user has authority to a secure token when passed to a callable service
  - Prevent duplicate tokens in the **CKDS** and **PKDS**
  - Raise the level of access authority required to create, write, and delete key labels
  - Raise the level of access authority required to export a token using the Symmetric Key Export callable service
  - Set additional restrictions on how keys can be used

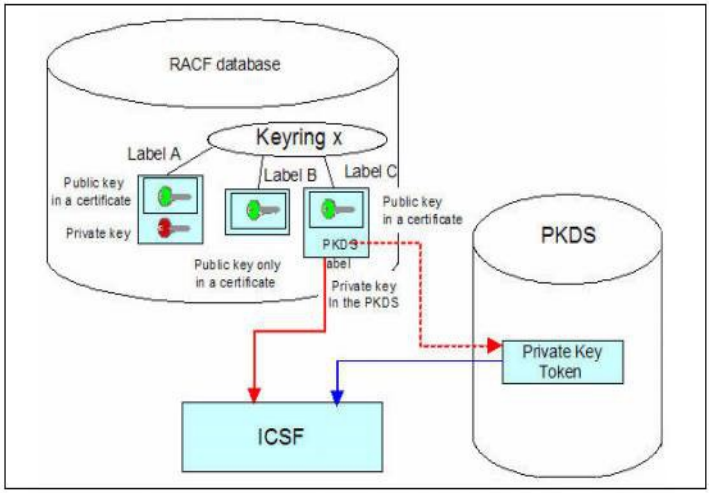
# Trusted Key Entry (TKE) Workstation

- **Components:**
  - Workstation with a 4765 Cryptographic Coprocessor
  - TKE 7.2 LIC
  - Smart card readers and smart cards
    - Required if using Enterprise PKCS #11 LIC
    - Optional if using IBM CCA LIC
- **Purpose:**
  - Used to manage multiple Cryptographic Coprocessors and keys on various generations of System z from a single point of control
    - Support requirements for standards
    - Simplification of tasks



# Putting it all together: ICSF Usage – RACF Digital Certificates

- RACF can utilize ICSF protected keys in it's digital certificate support to generate a certificate signed with an ICSF protected key:
  - RACDCERT GENCERT SUBJECTSDN(CN('cert01')) size(1024) withlabel('cert01') ID(user01) SIGNWITH(label('My CA'))
- **RACDCERT Command:**
  - Generates the encoded TBS (to be signed) certificate
  - Retrieves the private key:
    - Private key is stored in ICSF – RACF only has the ICSF private key label
  - Calls ICSF digital signature generate function (CSNDDSG) passing the TBS Cert and key label
- **ICSF:**
  - Checks callers authority: CSFKEYS, CSFSERV classes
  - Retrieves encrypted operational key from ICSF PKDS
  - Calls into coprocessor passing encrypted key and TBS cert
- **Coprocessor:**
  - Check if ACP is enabled for DSG
  - Decrypts operational key with PKDS master key
  - Performs digital signature generate and returns to caller
- **RACDCERT:**
  - Adds the returned signature to the encoded digital certificate

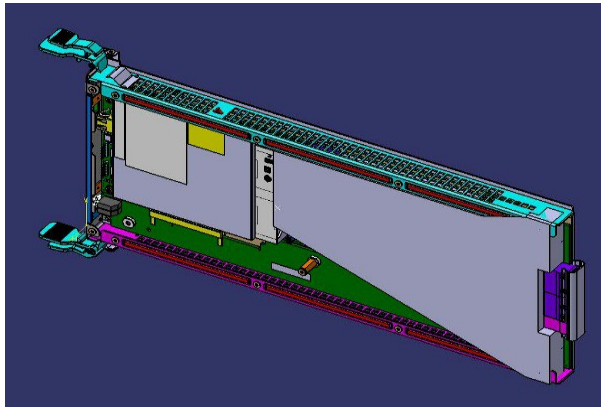


# Crypto Enhancements

# Common Cryptographic Architecture (CCA) Enhancements

- **VISA Format Preserving Encryption (VFPE)**
  - Support for VISA Format Preserving Encryption (VFPE) algorithms in CCA-based callable services. This support will rely on the Crypto Express5S coprocessor.
  - Format Preserving Encryption (FPE) refers to a method of encrypting data such that the resulting cipher-text has the same format and length as the input-clear text. This helps allow legacy databases to contain encrypted data of sensitive fields without having to restructure the database or applications.
  - Supported are functions for the VISA Data Secure Platform (Visa DSP) with Point to Point Encryption technology. Three new Visa DSP-related callable services are added to the CCA API. In addition to VFPE, support for the Visa DSP standard TDES encryption method is also available.
- **Greater than 16 Domain support**
  - Support to allow a cryptographic coprocessor to be shared across more than 16 domains, up to the maximum number of LPARs on the system.
  - This support relies on enhanced firmware available with a minimum microcode level for the Crypto Express4S and Crypto Express5S coprocessors. With the adjunct processor (AP) extended addressing (APXA) facility installed, the z Systems crypto architecture can support greater than 16 domains in an AP.
  - Customers will have the flexibility of mapping individual LPARs to unique crypto domains or continuing to share crypto domains across LPARs.

# Crypto Hardware Enhancements



- **Crypto Express5S:**
  - **Hardware improvements:**
    - Designed to offer better performance
  - **System z Partitions:**
    - Increase domains to support up to 85 LPARs
  - **RAS Improvements**
  - **Cryptographic Enhancements:**
    - Support for protocols and algorithms **in hardware** for better crypto performance

- **z13 CPACF:**
  - Optimized for better performance

# More Crypto on z/OS

# Certificate Authority: PKI Services

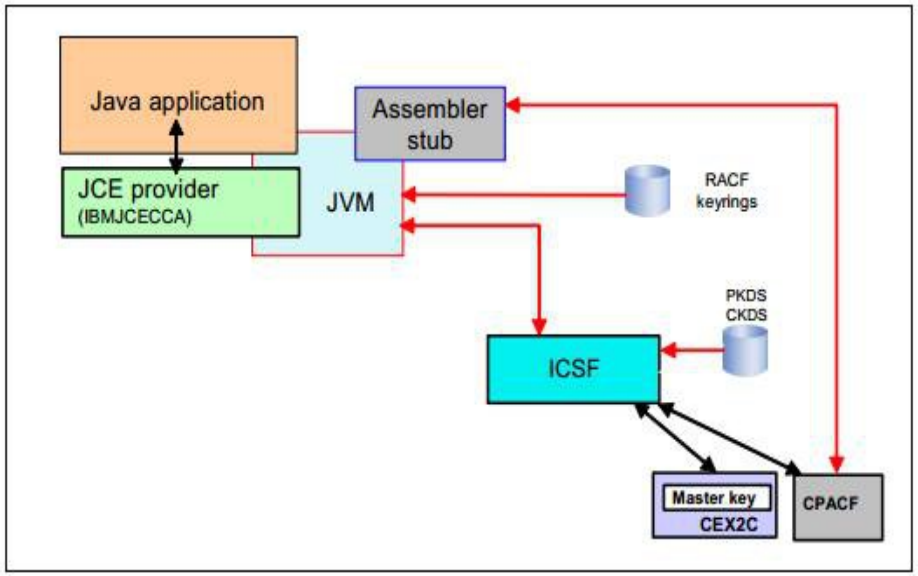
- **PKI Services** provides full certificate life cycle management
- **Request**, **create**, **renew**, **revoke** certificates
  - Provides certificate status:
    - **Certificate Revocation List** (CRL)
    - **Online Certificate Status Protocol** (OCSP)
  - Generation and administration of certificates via customizable web pages
  - Support **Simple Certificate Enrollment Protocol (SCEP)** for routers to request certificates automatically
  - **Automatic notifications** or renewal of expiring certificates

# System SSL

- An element of the z/OS base Cryptographic Services element provides:
  - **A certificate management utility**, gskkyman, for managing certificates within a key database file as well as a suite of APIs to allow application writers the ability to write their own certificate management programs.
  - Provides a mechanism (suite of C/C++ POSIX callable application programming interfaces (APIS)) for applications to securely communicate over an open communications network using **SSL/TLS protocol**
  - Although not part of the SSL protocol support, System SSL also contains a suite of APIs that allows for applications to **build/read PKCS#7 messages**.

# Java Cryptography in z/OS

- **Java Crypto Extension (JCE) - IBMJCECCA**
  - Provides framework and implementation for cryptographic services in java:
    - Symmetric Algorithms
    - Asymmetric
    - Key Generation
    - Key agreement
    - MAC Algorithms
    - RNG
    - RACF Key Rings
    - Much more
  - Utility:
    - keytool




- **More info:**
  - http://www-03.ibm.com/systems/z/os/zos/tools/java/products/j5jcecca.html

# Reference

- **ICSF Publications:**
  - SA22-7520 ICSF Systems Programmer's Guide
  - SA22-7521 ICSF Administration Guide
  - SA22-7522 ICSF Application Programmer's Guide
- **Java Crypto:**
  - http://www-03.ibm.com/systems/z/os/zos/tools/java/products/j5jcecca.html
- **PKI Services:**
  - http://www-03.ibm.com/systems/z/os/zos/features/pki/
- **TechDocs:**
  - www.ibm.com/support/techdocs
- **z/OS Web Download Site:**
  - http://www.ibm.com/systems/z/os/zos/tools/downloads/index.html

# Intro to z/OS Crypto and ICSF

*Ross Cooper, CISSP®*
*IBM Corporation*

*March 2nd, 2015*
*Session: 16777*

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides **education**, professional **networking** and industry **influence**.