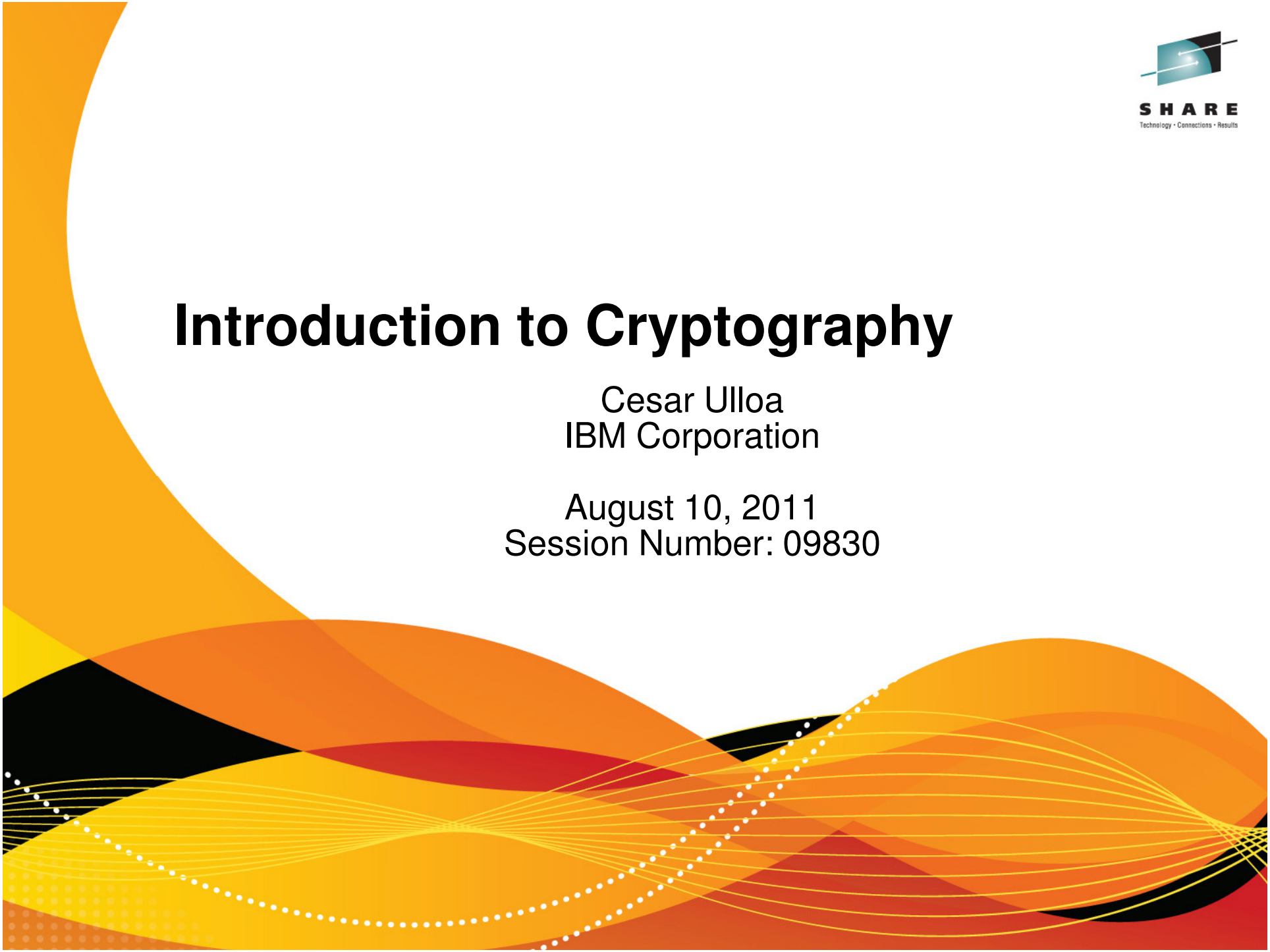


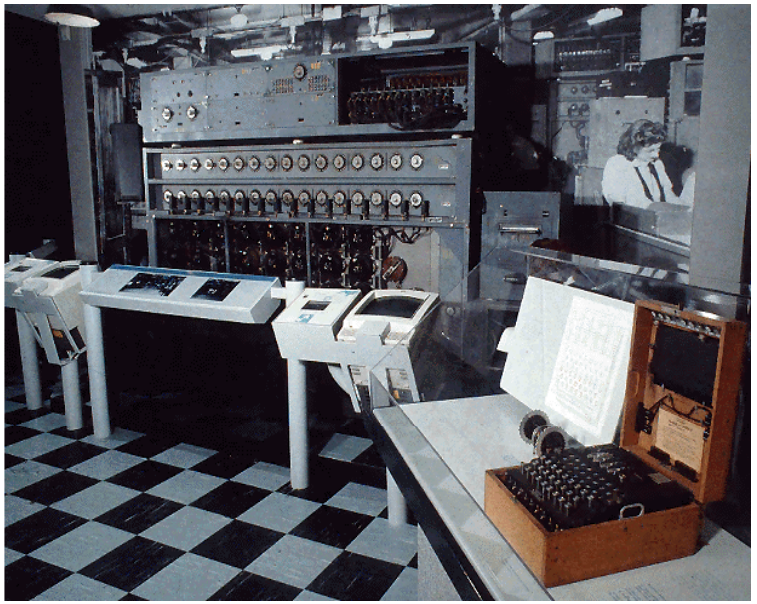
Introduction to Cryptography

Cesar Ulloa
IBM Corporation

August 10, 2011
Session Number: 09830

Agenda – Intro To Crypto

- Some background
  - Laws & Regulations
  - Crypto Standards
- Crypto Functions
- Crypto Hardware
- Keys
  - Master Keys
  - Operational Keys
- ICSF & Products that use crypto
- Linux



What is Cryptography?

Cryptography (or cryptology; from Greek κρυπτός, *kryptos*, "hidden, secret"; and γράφω, *gráphō*, "I write", or -λογία, *-logia*, respectively)**[1]** is the practice and study of hiding information. In modern times cryptography is considered a branch of both mathematics and computer science and is affiliated closely with information theory, computer security and engineering.

From Wikipedia

Crypto Functions

- Data Confidentiality
- Data Integrity
  - Modification Detection
  - Message Authentication
  - Non-repudiation
- Financial Functions
- Key Security & Integrity

Laws and Regs

- Health Insurance Portability and Accountability Act of 1996 (HIPAA)
- California Senate Bill 1386
- Gramm-Leach-Bliley Act (GLBA)
- *Sarbanes-Oxley (SOX)*
- Payment Card Industry Standards (PCI)

Cryptographic Standards

- CCA (Comon Crpytographic Architecture)
- PKCS (Public-Key Cryptography Standard)
- INTEL CDSA (Common Data Security Architecture)
- OCSF (Open Cryptographic Services)
- ANSI (American National Standards Association)
- ISO (International Organization for Standardization)
- FIPS (Federal Information Processing Standards)

Kerckhoff's Principle

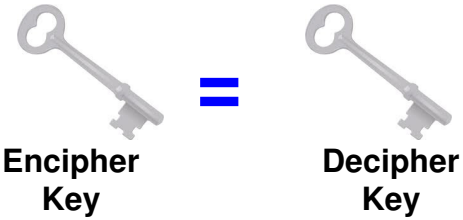
- "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge."

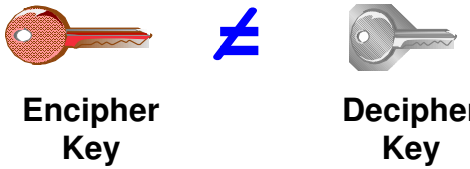
One-Time Pads

- Generate a random key of equal length to you message then exclusive-or (XOR) the key with your message.
- This is information theoretically secure …but:
  - “To transmit a large secret message, first transmit a large secret message”
  - One time means one time.
  - Need to transmit a key per message per recipient
  - Keys are as big as messages.

Methods of Encryption

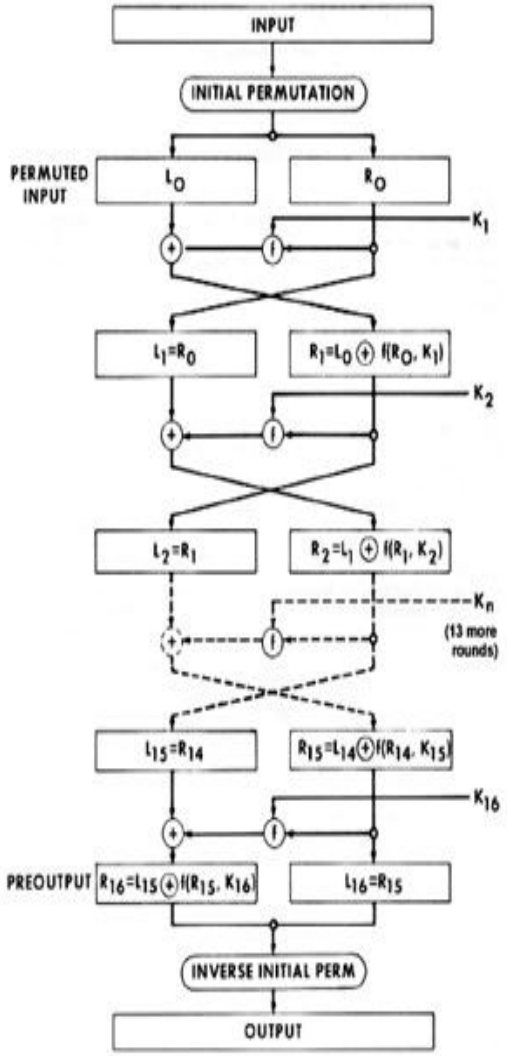
- Symmetric Cryptography
  - DES/TDES
  - AES
  - Blowfish
  - IDEA
  - RC4, RC5, RC6
- Asymmetric Cryptography
  - RSA
  - Diffie-Hellman
  - ECC



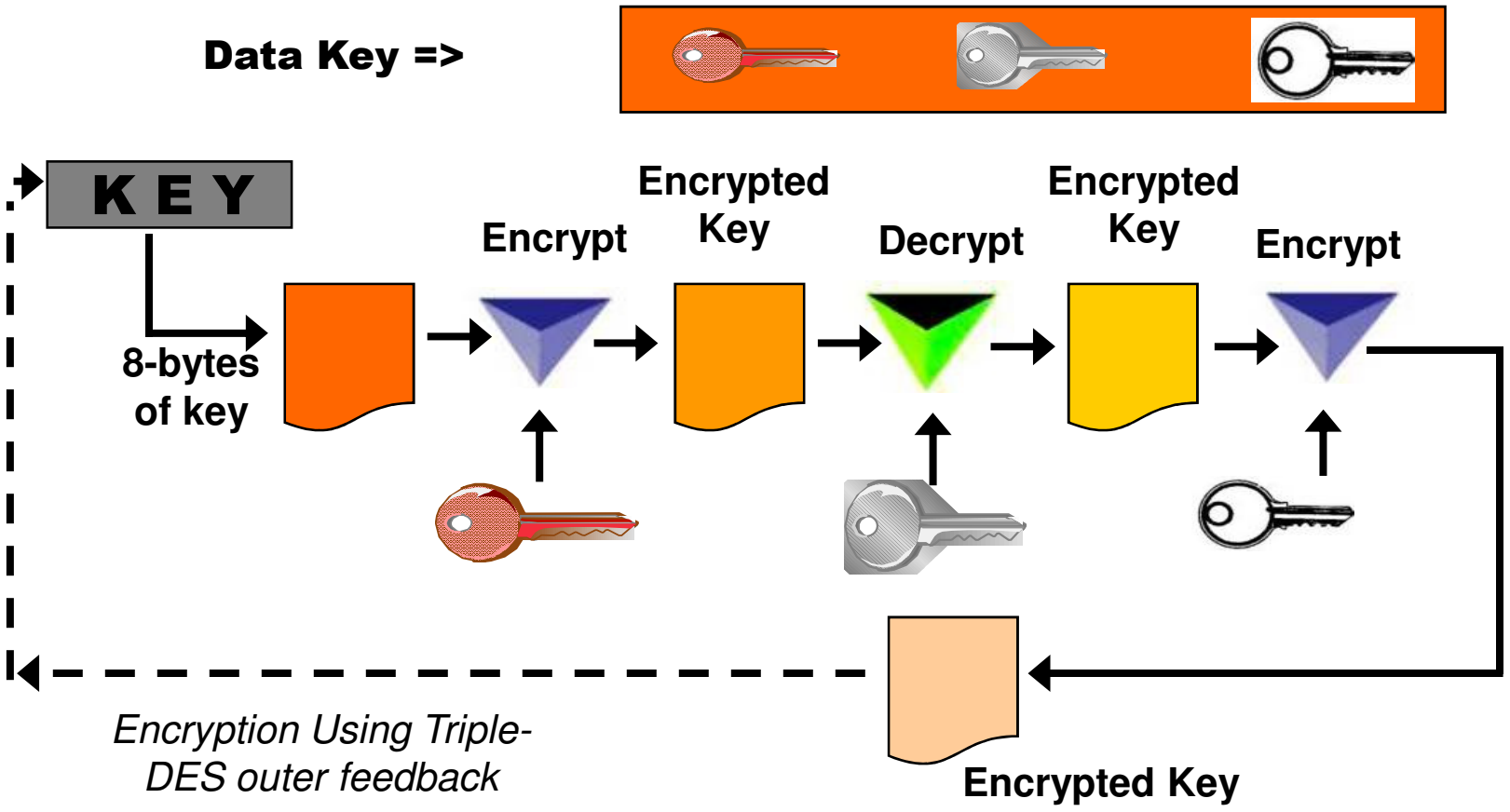


DES and TDES

- Data Encryption Standard (DES): A strong standardized 8-byte, 56-bit (with parity bits) cipher designed for modern computers.
- Originally designed by IBM and called “lucifer”. Tweaked by NSA and published in 1975.
- In 1999, a DES key was brute forced in 24 hours
- Triple DES (TDES/3DES): Effectively 24-byte, 112-bit cipher. Still in use.



Data Confidentiality – DES/TDES



Data Confidentiality - AES

- Rijndael Algorithm
  - Block Cipher (16-byte blocks)
  - 128-, 192-, 256-bit key length
  - Multiple round
  - Four steps per round (Byte substitution, shift row, mix column, add round key)

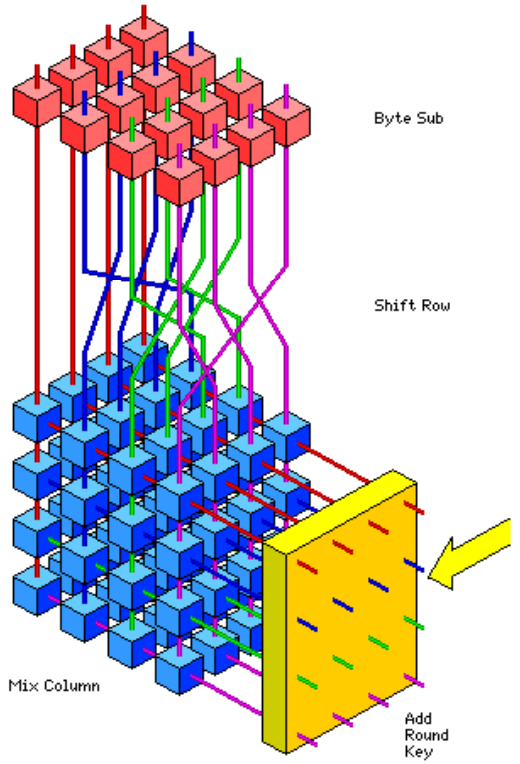
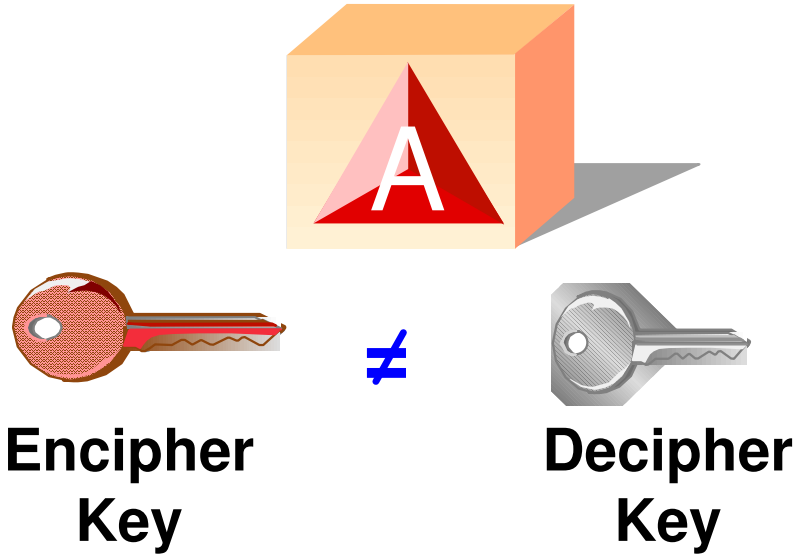


Image from http://www.esat.kuleuven.ac.be/~rijmen/rijndael

Public Key Architecture - PKA

- Asymmetric Keys
  - RSA
  - Diffie-Hellman
  - Elliptic Curve



RSA Public Key Cryptography

- Generate 2 prime numbers (each over 100 digits long)
- Multiply primes to get modulus, N

  N = 7 x 17 = 119
- Select odd number, E, that will be the second part of the public key
- Public Key (N E)
- Compute second part of private key, D

  (P-1) x (Q-1) x (E-1)

  Add 1 to result

  Divide by E to get D
- Private Key (N D)

P=7 Q=17

N = 7 x 17 = 119

E = 5

119 5 <= Public Key

(7-1) x (17-1) x (5-1) = 384

384 + 1 = 385

D = 385/5 = 77

119 77 <= Private Key

Encipher Message – 'SELL'

- P = 7; Q = 17; N = 119; E = 5; D = 77
- Public Key (N E)
- Private Key (N D)
- Convert characters to numerics
  - e.g. A=1, B=2, C=3 ….
  - Plaintext 'SELL' becomes 19 5 12 12
- Raise that character value to power E
- Divide by first part of Public Key
- And get the remainder
- Ciphertext

119 5

119 77

'S' => 19

('S' => 19**5 => 2476099)

2476099 / 119 = 20807

e_{PK}('S') = 66

66 31 3 3

Decipher Message – 'SELL'

- P = 7; Q = 17; N = 119; E = 5; D = 77
- Public Key (N E)
- Private Key (N D)

119 5

119 77

- Raise ciphertext (66 31 3 3) to power of D
- Divide result by modulus N and get the remainder
- Remainder is numeric equivalent of the character sent
- Plaintext

66**77 = 1273….

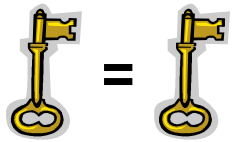
1273…. / 119 = 1069 Mod 19

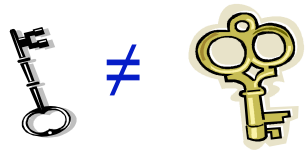
19 = 'S'

19 5 12 12 or 'S E L L'

Why Asymmetric and Symmetric Keys?

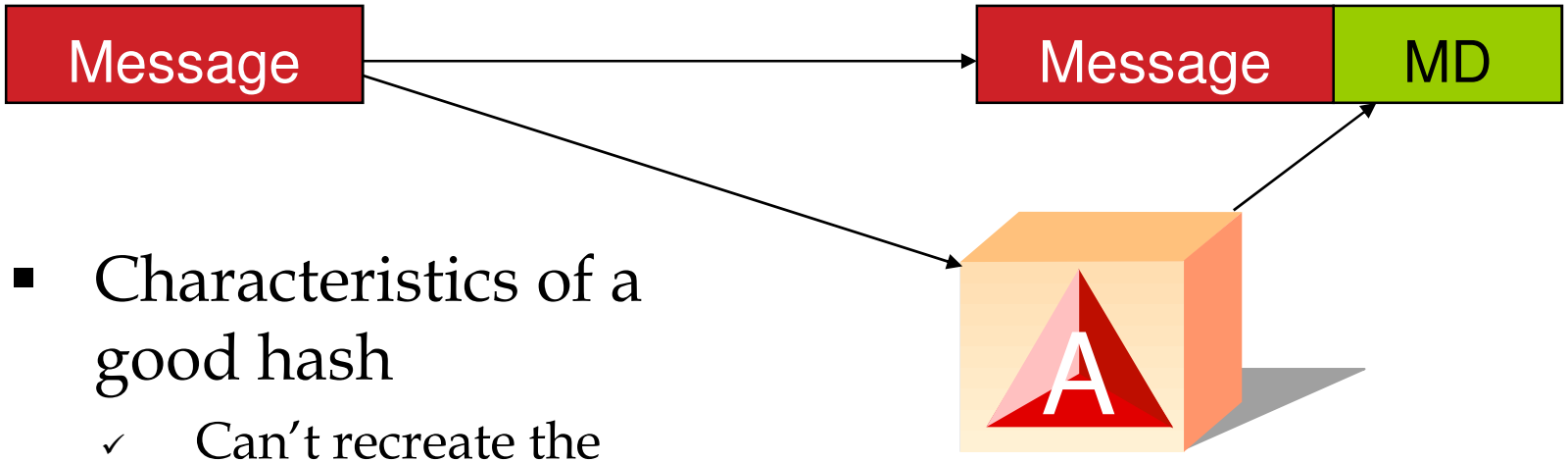
- Symmetric
  - Pros – less resource intensive
  - Cons – requires key to be shared securely
- Asymmetric
  - Pros – it's strength, can be used to establish a secret between two parties
  - Cons – expensive, in terms of performance





Data Integrity – Modification Detection

Has the message changed?



- Characteristics of a good hash
  - ✓ Can't recreate the message from the hash
  - ✓ Two different messages are statistically unlikely to generate the same hash value

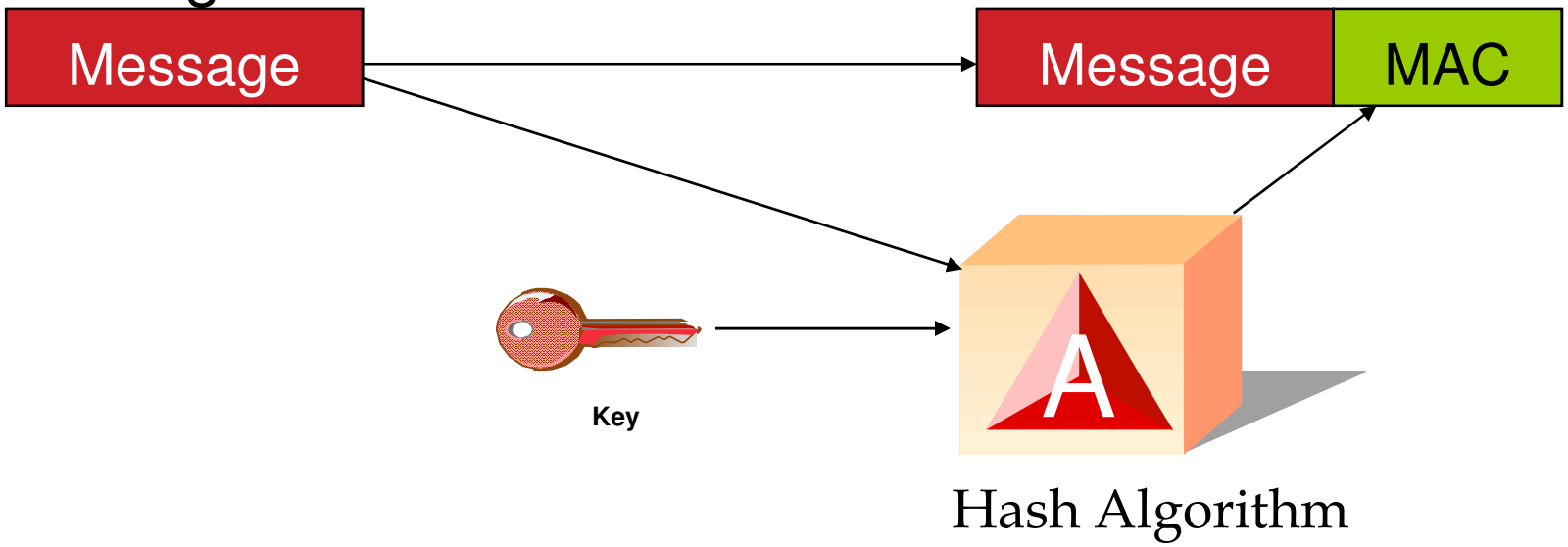
Hashing Example

SHA1("The quick brown fox jumps over the lazy dog")
= 2fd4e1c6 7a2d28fc ed849ee1 bb76e739 1b93eb12

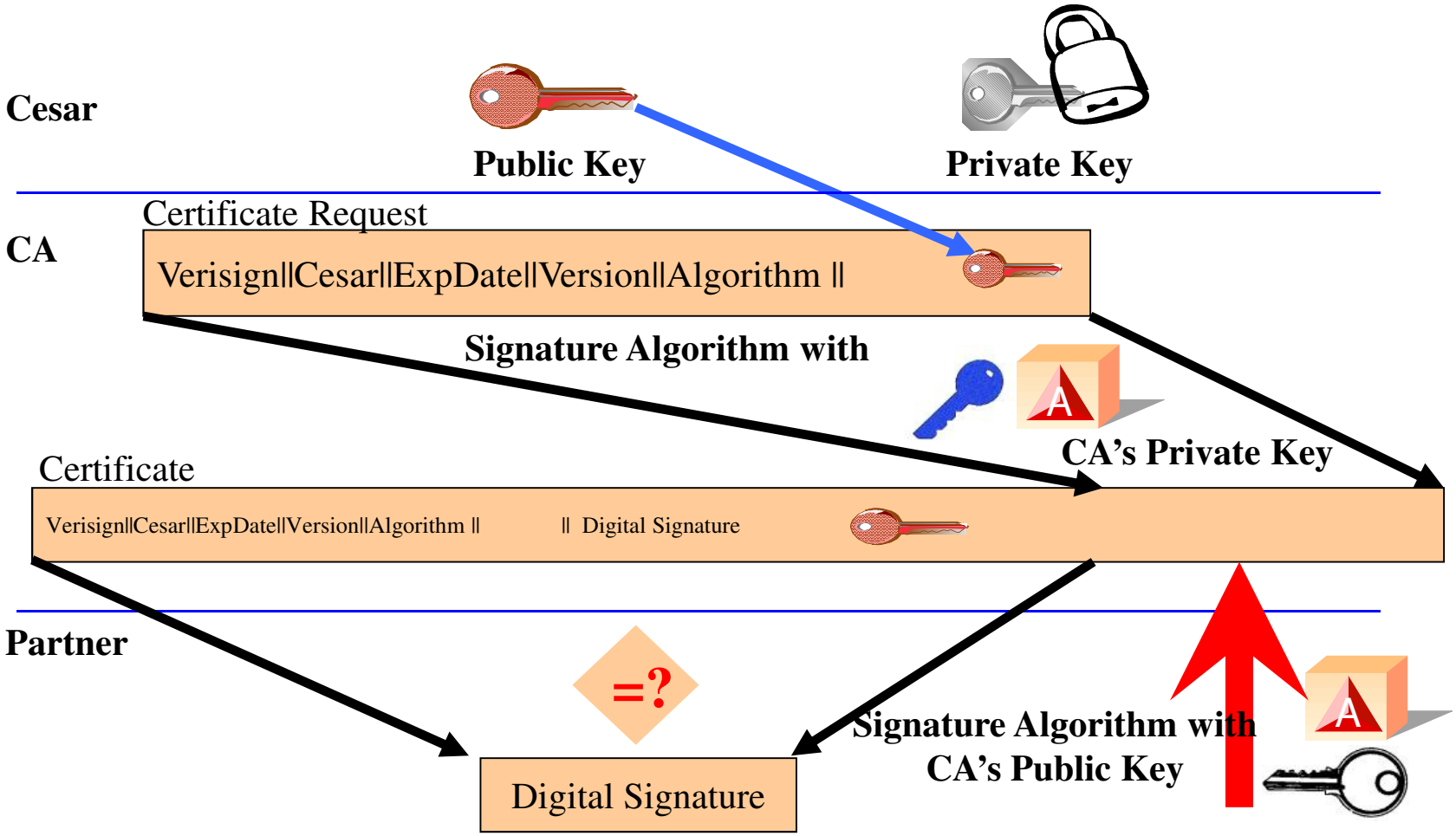
SHA1("The quick brown fox jumps over the lazy **c**og")
= de9f2c7f d25e1b3a fad3e85a 0bd17d9b 100db4b3

Data Integrity – Modification Authentication

Has the message changed? And who sent the message?

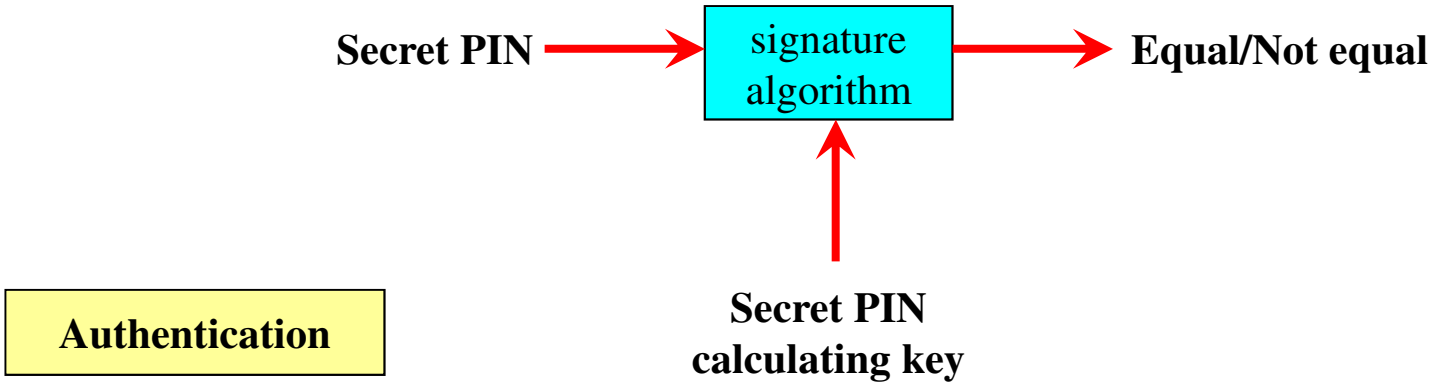


Data Integrity – Digital Certificates



Financial Services

- PIN Generation
- PIN Verification
- PIN Export/Import



Suite B

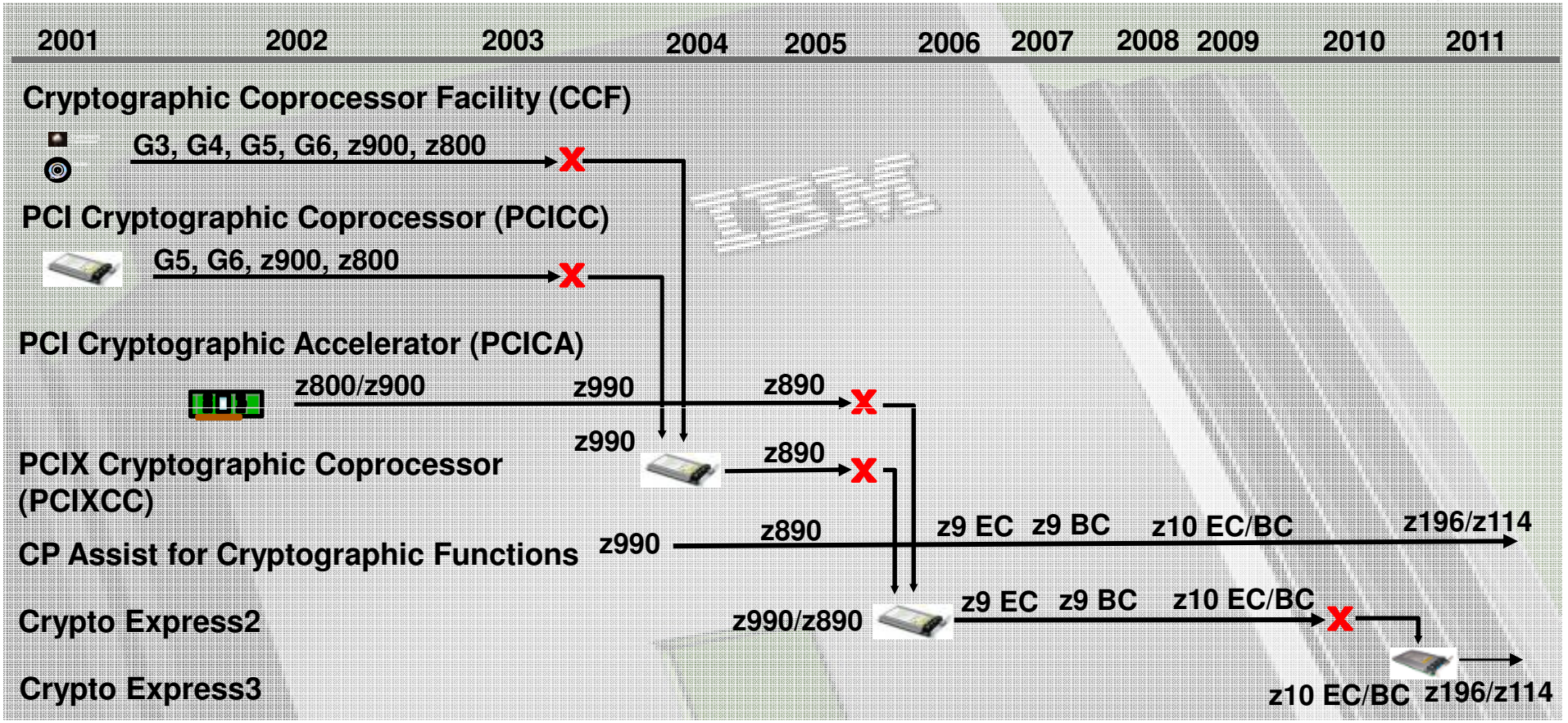
- Symmetric Encryption
  - AES w/key sizes of 128 and 256
- Digital Signatures
  - EC DSA
- Key Agreement
  - EC Diffie-Hellman
- Message Digest
  - SHA-2 (SHA-256 and SHA-384)

http://www.nsa.gov/ia/programs/suiteb_cryptography/

Clear Key / Secure Key / Protected Key

- Clear Key – key may be in the clear, at least briefly, somewhere in the environment
- Secure Key – key value does not exist in the clear outside of the HSM (secure, tamper-resistant boundary of the card)
- Protected Key – key value does not exist outside of physical hardware, although the hardware may not be tamper-resistant

System z Crypto History



- Cryptographic Coprocessor Facility – Supports "Secure key" cryptographic processing
- PCICC Feature – Supports "Secure key" cryptographic processing
- PCICA Feature – Supports "Clear key" SSL acceleration
- PCIXCC Feature – Supports "Secure key" cryptographic processing
- CP Assist for Cryptographic Function allows limited "Clear key" crypto functions from any CP/IFL
  - NOT equivalent to CCF on older machines in function or Crypto Express2 capability
- Crypto Express2 – Combines function and performance of PCICA and PCICC
- Crypto Express3 – PCI-e Interface, additional processing capacity with improved RAS

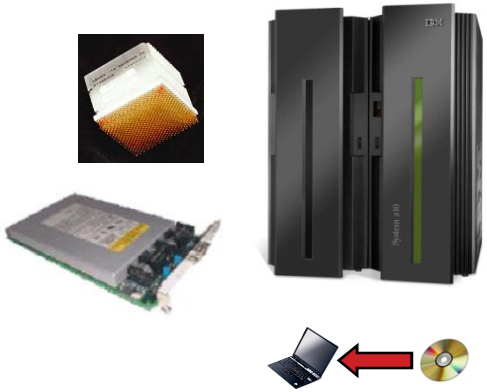
System z Clear Key Cryptographic Hardware – z9 (EC & BC), z10 (EC (GA3) & BC (GA2)), z196 & z114

- CP Assist for Cryptographic Function (CPACF)
  - DES (56-, 112-, 168-bit), new chaining options
  - AES (128-, -192, 256-bit), new chaining options
  - SHA-1, SHA-256, SHA-512 (SHA-2)
  - PRNG
  - Protected Key

TechDoc WP100810 – A Synopsis of System z Crypto Hardware

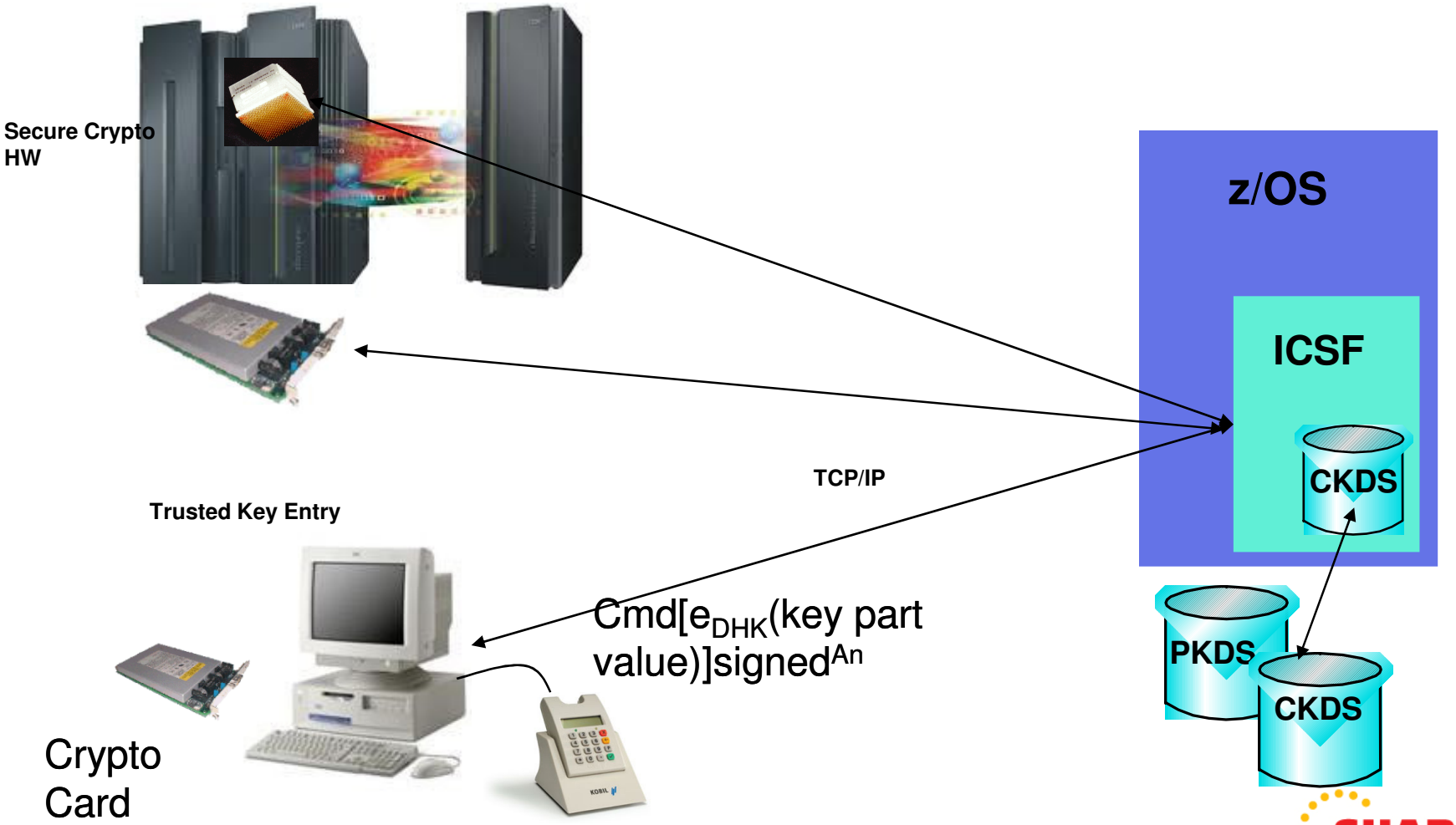
System z Secure Key Crypto Hardware - CEX2/CEX2-1P and CEX3/CEX3-1P

- Secure Key DES/TDES
- Secure Key AES
- Financial (PIN) functions
- Key generation / Key management
- Random Number Generate and Generate Long
- Protected Key Support
- SSL Handshakes, ECDSA support



TechDoc WP100810 – A Synopsis of System z Crypto Hardware

TKE – Trusted Key Entry Workstation



Master Keys

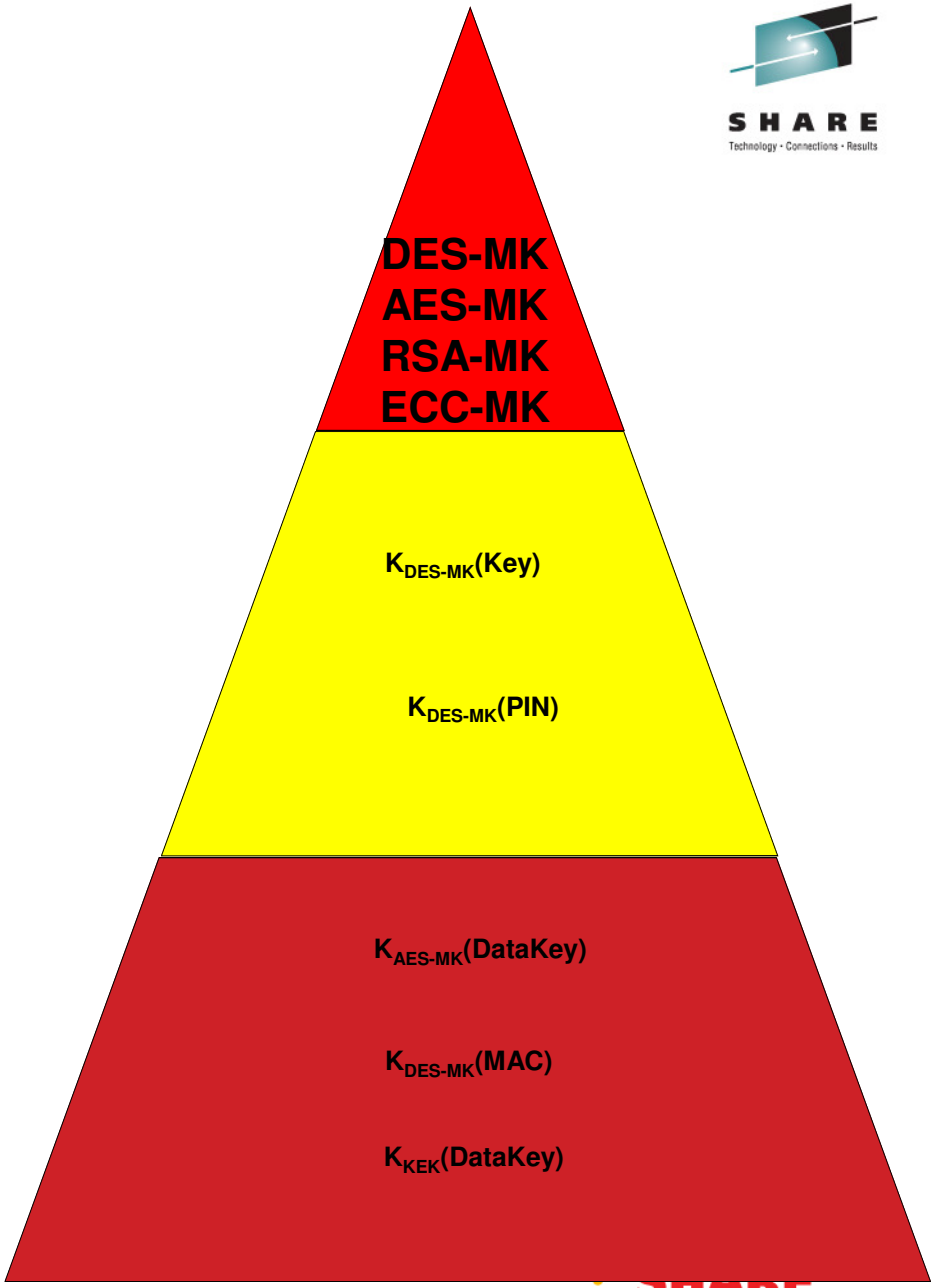
- ICSF uses four master keys to protect operational keys
  1. **DES Master Key (DES-MK)**
     - 128 bit key
     - Protects DES/TDES (symmetric) application keys
  2. **AES Master Key (AES-MK)**
     - 256 bit key
     - Protects AES (symmetric) application keys
  3. **RSA Master Key (RSA-MK)**
     - 192 bit key
     - Protects RSA (asymmetric) private keys
  4. **Elliptic Curve Master Key (ECC-MK)**
     - 256 bit key
     - Protects ECC (asymmetric) private keys

- Stored within the secure hardware boundary of the crypto coprocessor

Key Hierarchy

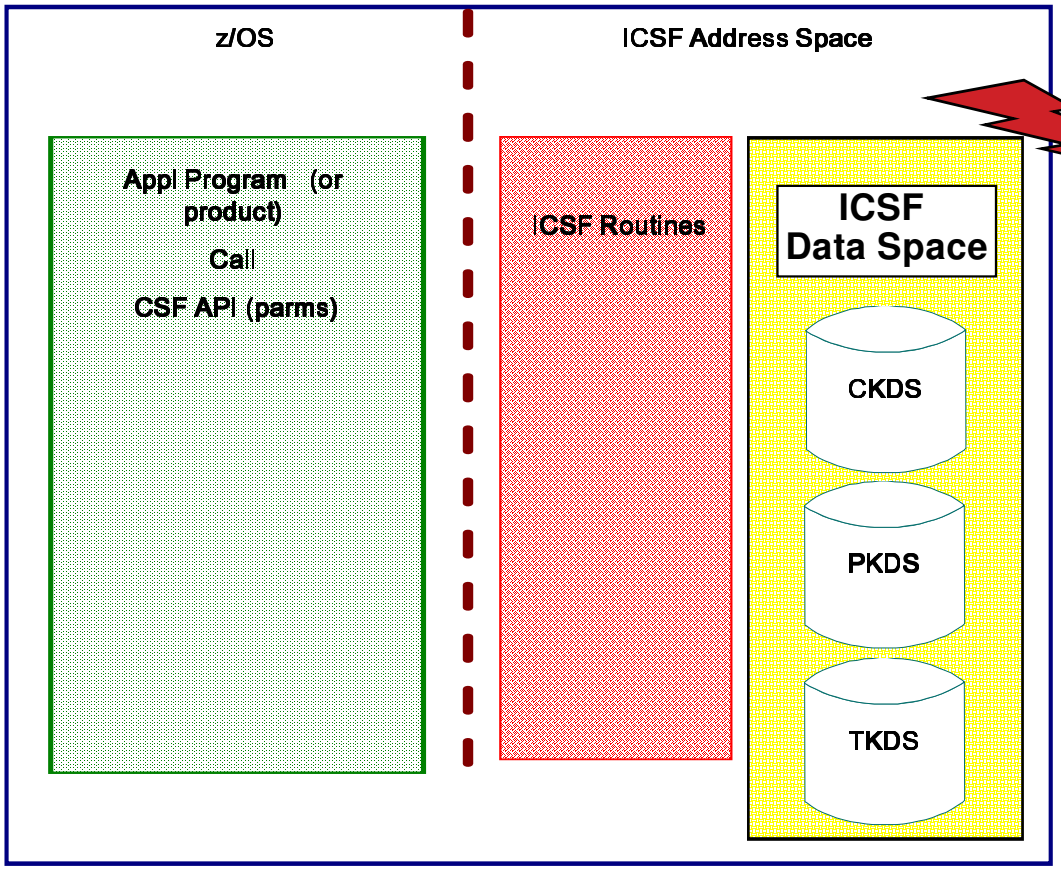
- Master Keys – clear value resides only inside the secure hardware
- Key-Encrypting-Keys
- Operational Keys

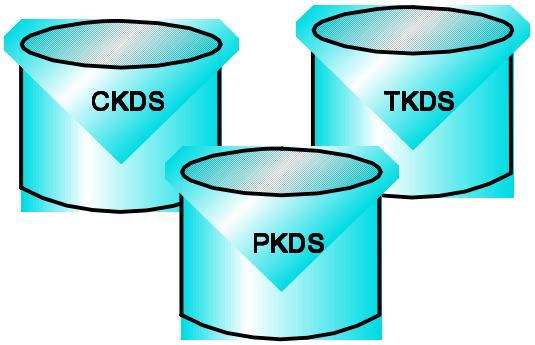
Data	Dataxlat
MAC/MACVER	PIN
DATAM/DATAMV	System
Exporter/Importer	IMP-PKA



ICSF – Interface to the hardware

- APIs
- Key Storage
- Load Balancing
- Security

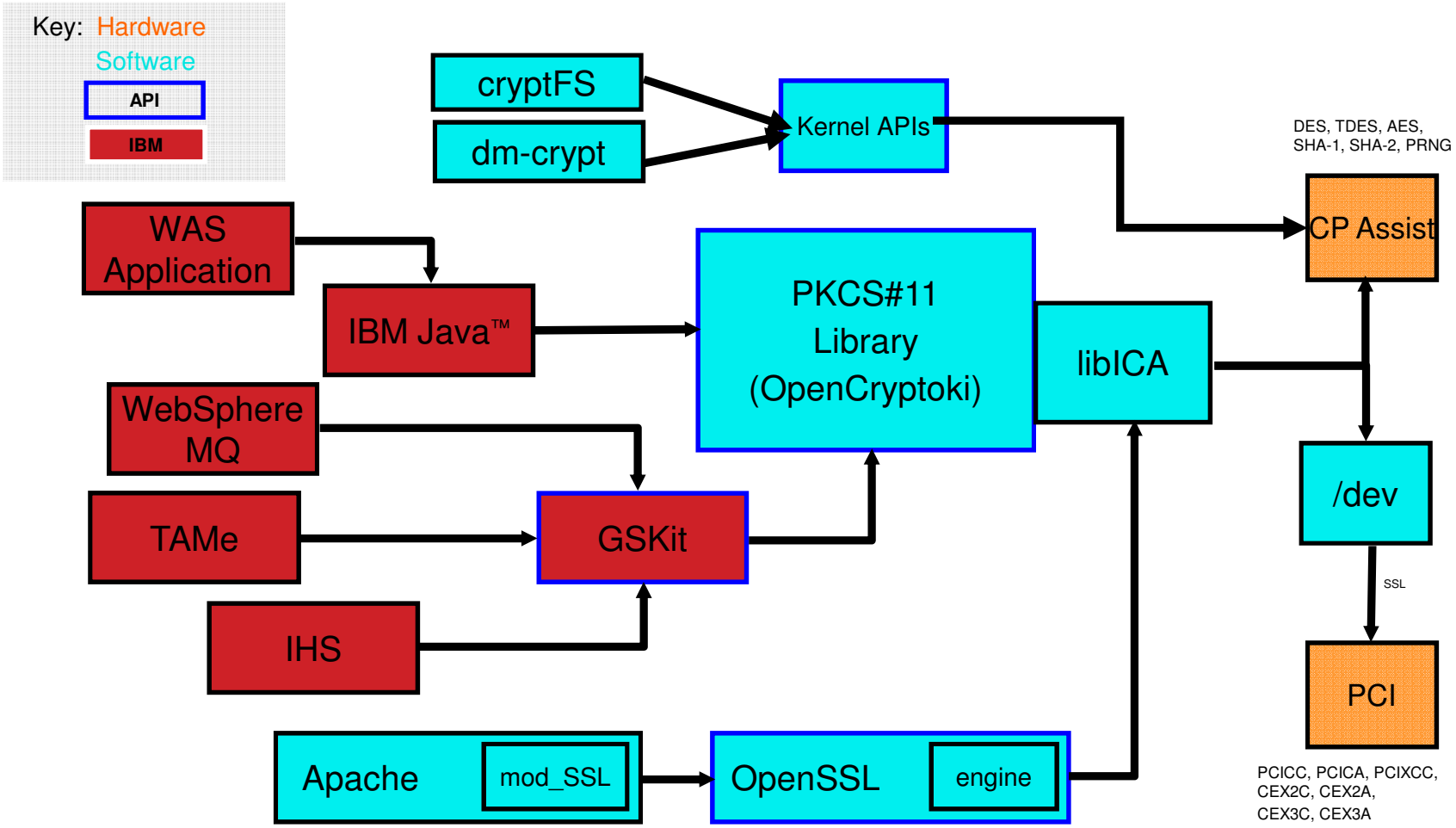




Products that use the crypto infrastructure

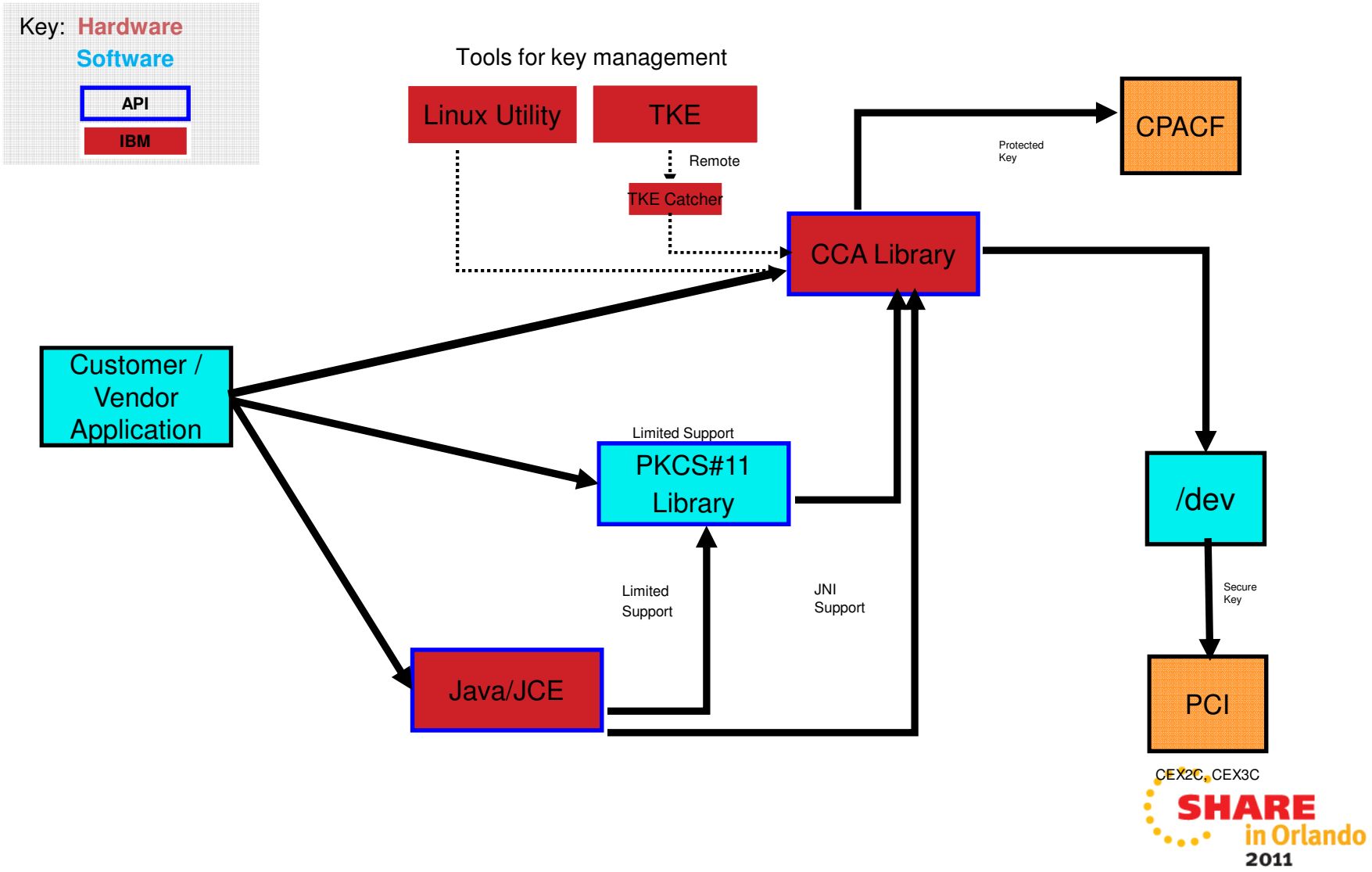
- Data Encryption Tool for IMS and DB2 Databases
- DB2 Built-In Functions
- Encryption Facility for z/OS
- TKLM / ISKLM(used to be EKM)
- System SSL
- IPSEC
- PKI Services

Linux on System z Clear Key



OpenCryptoki: http://www.ibm.com/developerworks/linux/library/s-pkcs/

Linux on System z – Secure Key



References

- Cryptography Books
  - Bruce Schneier, “Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in ‘C’”, Addison Weley Longman, Inc. 1997
  - Simon Singh, “The Code Book”, Anchor Books, 1999
  - Niels Ferguson, Bruce Schneier, “Practical Cryptography”, Wiley Publishing, Inc. 2003
- Standards
  - www.ietf.org – Internet Engineering Task Force
  - http://csrc.nist.gov/ – Computer Security Resource Center of NIST
  - http://www.rsa.com/rsalabs/ - Research site for RSA Security
- Magazines and Newsletters
  - www.scmagazine.com/ – SC Magazine
  - www.counterpane.com – Bruce Schneier web site with monthly newsletter

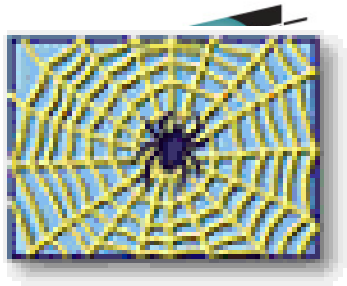
IBM Pubs

- ICSF Overview, SA22-7519
- ICSF Administrator's Guide, SA22-[illegible]
- ICSF Application Programmer's Guide, SA22-7522
- ICSF System Programmer's Guide, SA22-7520

z/OS Web Download Site

- www.ibm.com/systems/z/os/zos/downloads/

IBM Resources (on the web)



- Redbooks – www.redbooks.ibm.com (search on ‘crypto’)
  - IBM zEnterprise 196 Configuration Setup, SG24-7834
  - IBM zEnterprise System Technical Introduction, SG24-7832
  - IBM zEnterprise System Technical Guide, SG24-7833
  - IBM System z196 Enterprise Class Technical Guide, SG24-7516
- ATS TechDocs Website – www.ibm.com/support/techdocs (search on ‘crypto’)
  - WP100810 – A Synopsis of System z Crypto Hardware
  - WP100647 – A Clear Key / Secure Key /Protected Key Primer

Java Crypto Resources

- Java Security Page
  - **ibm.com**/developerworks/java/jdk/security
- How to use the IBM Java Hardware Crypto Providers
  - **ibm.com**/developerworks/java/jdk/security/142/HardwareCryptoHow-to.html
- The IBMPKCS11Impl Provider Guide
  - **ibm.com**/developerworks/java/jdk/security/50/secguides/pkcs11implDocs/IBMJavaPKCS11ImplementationProvider.html
- IBM Java PKCS#11 Supported Devices
  - **ibm.com**/developerworks/java/jdk/security/50/secguides/pkcs11implDocs/IBMPKCS11SupportList.html
- CCA Resources
  - General Secure Key Overview
    http://www-03.ibm.com/security/cryptocards/pciecc/overview.shtml
  - Download page for CCA library for S390 Linux
    http://www-03.ibm.com/security/cryptocards/pciecc/ordersoftware.shtml
  - Documentation download page
    http://www-03.ibm.com/security/cryptocards/pciecc/library.shtml

Secure Key Crypto – Information & Download

- Crypto Card – CryptoExpress3
  - **ibm.com**/security/cryptocards/pciecc/overview.shtml
  - Programmer's Guide
    ibm.com/security/cryptocards/pciecc/library.shtml
  - CCA Library Download
    ibm.com/security/cryptocards/pciecc/ordersoftware.shtml
- Crypto Card – CryptoExpress2
  - **ibm.com**/security/cryptocards/pcixcc/overview.shtml
  - Programmer's Guide
    ibm.com/security/cryptocards/pcixcc/library.shtml
  - CCA Library Download
    ibm.com/security/cryptocards/pcixcc/ordersoftware.shtml
- Current Solution Brief
  - ftp://public.dhe.ibm.com/common/ssi/pm/sp/n/zss03052usen/ZSS03052USEN.PDF

Questions …