Intro To Crypto On System z
Share 12687
San Francisco, CA February, 2013

Greg Boyd (boydg@us.ibm.com)

QR Code

Agenda – Intro to Crypto

- **Some background**
  - Laws & Regulations
  - Crypto Standards
- **Crypto Functions**
- **Crypto Hardware**
- **Master Keys**
- **ICSF**
- **Linux**

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)**

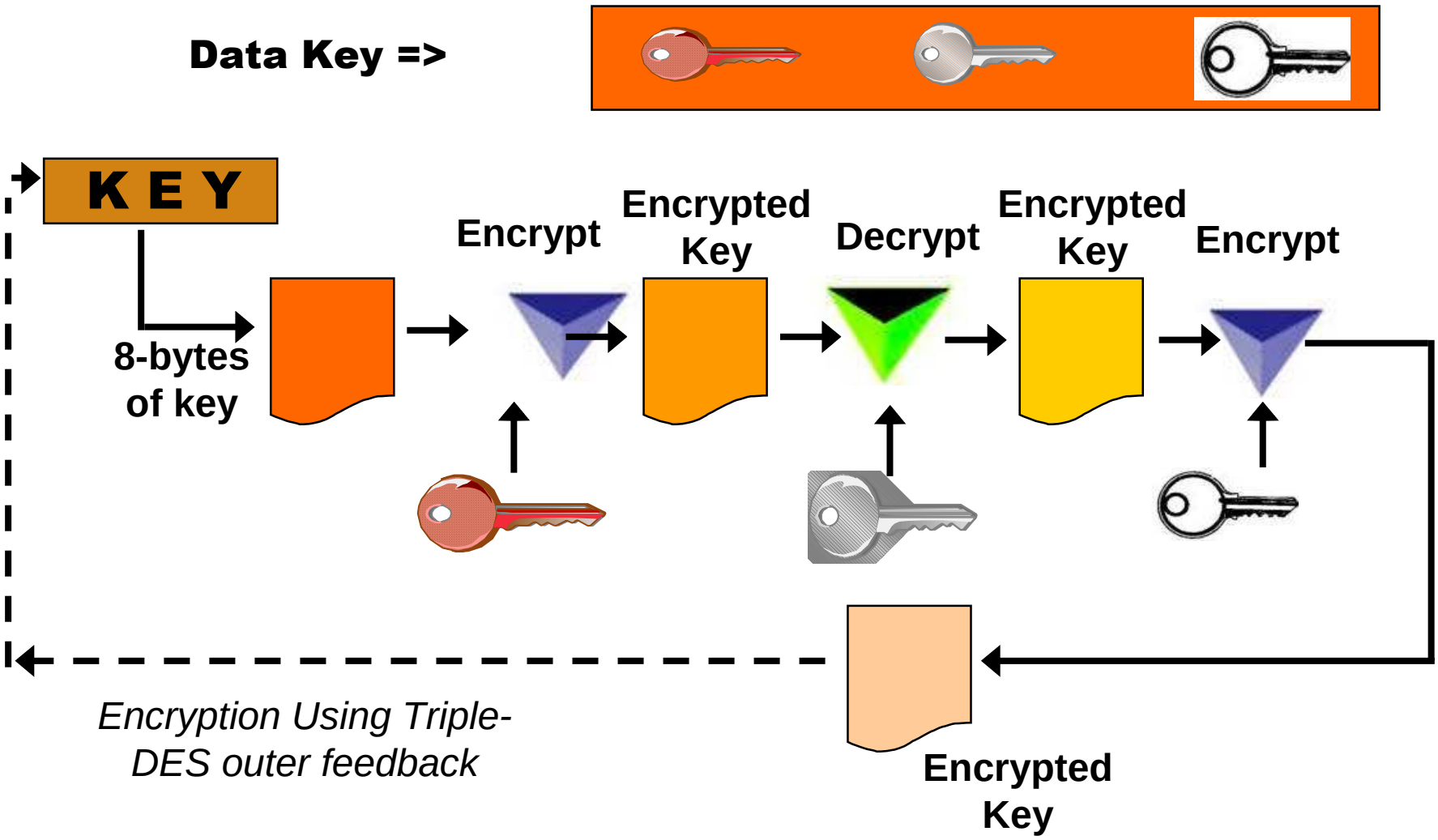
Cryptographic Standards

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

Crypto Functions

- **Data Confidentiality**
  - Symmetric – DES/TDES, AES
  - Asymmetric – RSA, Diffie-Hellman, ECC
- **Data Integrity**
  - Modification Detection
  - Message Authentication
  - Non-repudiation
- **Financial Functions**
- **Key Security & Integrity**

Data Confidentiality – DES/TDES



Data Confidentiality - AES

- **Rijndael Algorithm**
  - Block Cipher (16-byte blocks)
  - 128-, 192, 256-bit Key Length
  - Multiple Rounds
  - Four Steps per Round (Byte Substitution, Shift Row, Mix Column, Add Round Key)

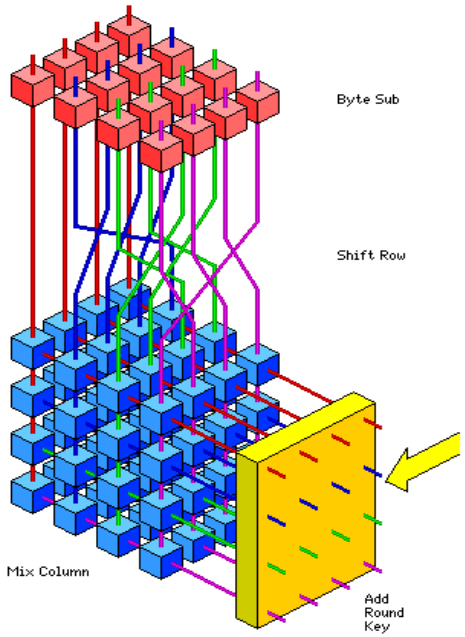
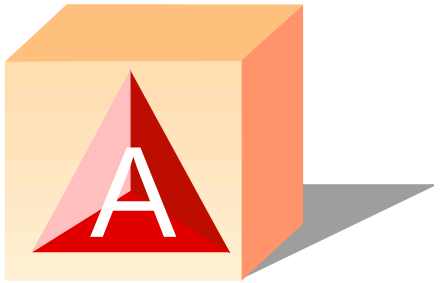
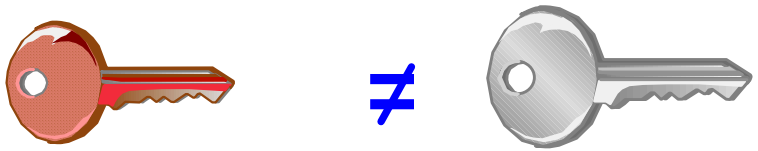


Image from http://www.esat.kuleuven.ac.be/~rijmen/rijndael/

Public Key Architecture - PKA

- **Asymmetric Keys**
  - RSA, Rivest Shamir and Adleman
  - Diffie-Hellman
  - Elliptic Curve (ECC)





Encipher Key ≠ **Decipher Key**

RSA Public Key Cryptography

▪ Generate 2 prime numbers	P = 7 Q = 17
(each over 100 digits long)	
▪ Multiply primes to get modulus, N	N = 7 x 17 = 119
▪ Select odd number, E, that will	E = 5
be the second part of the public key	
▪ **Public Key (N E) ➔**	**119 5**
▪ Compute second part of private key, D	
(P-1) x (Q-1) x (E-1)	(7-1) x (17-1) x (5-1) = 384
Add 1 to result	384 + 1 = 385
Divide by E to get D	D = 385/5 = 77
▪ **Private Key (N D) ➔**	**119 77**

Encipher Message ‘SELL’

- P = 7; Q = 17; N = 119; E = 5; D = 77

Public Key (N E) ➔ **119 5**

Private Key (N D) ➔ **119 77**

- Convert characters to numeric 19 5 12 12

E.g. a=1, b=2, c=3 ….

- Raise that character value to power E (‘S’ => 19**5 =>) 2476099
- Divide that by first part of Public Key to get the Modulus

2476099 MOD 119 = 66

$e_{PublicKey}$(S) => 66

- Ciphertext 66 31 3 3

Decipher Message ’66 31 3 3’

- P = 7; Q = 17; N = 119; E = 5; D = 77

Public Key (N E) ➔ **119 5**

Private Key (N D) ➔ **119 77**

- Raise ciphertext to power D — 66**77 = 1273160…..
- Divide result by first half of private key to get the modulus

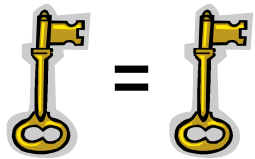
  1273160….. MOD 119 = 19
- Convert numeric back to character — 19 5 12 12

$d_{PrivateKey}$(66) => “S”

- Plaintext — 19 5 12 12 or C’S E L L’

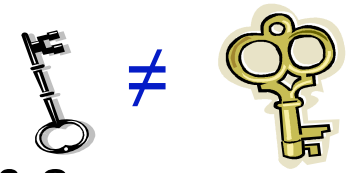
Why Symmetric and Asymmetric Keys?

- **Symmetric**



  - plus - less resource intensive
  - minus - requires key to be shared securely

- **Asymmetric**



  - plus - its strength, can be used to establish a secret between two parties
  - minus - expensive, regarding performance

Hash and Key Lengths

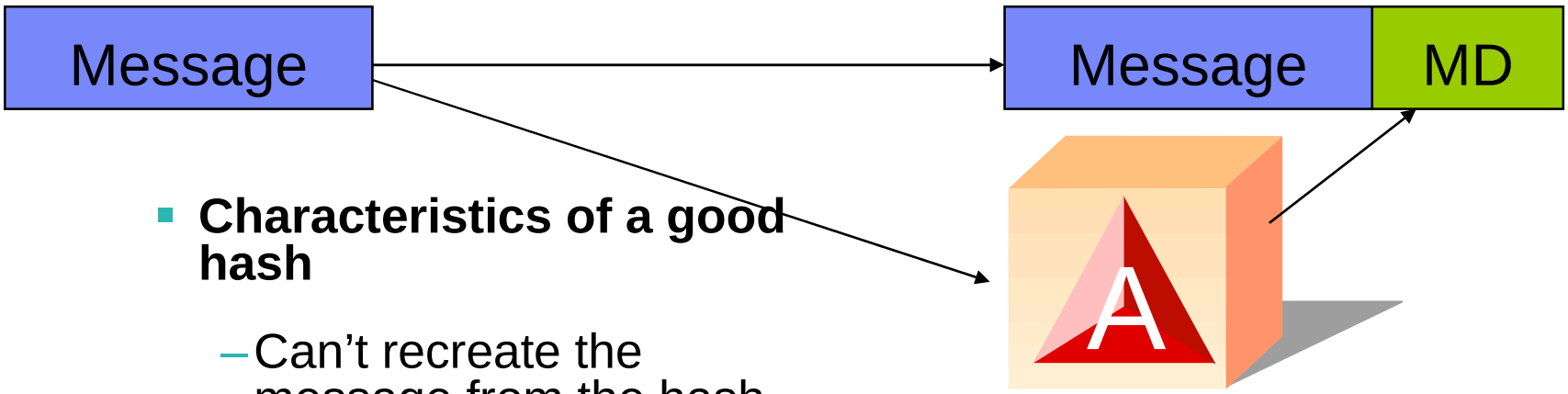
- **Why do we care about lengths a of keys and hashes?**
  - The longer a key the lower the probability to guess the right key
  - The longer a hash the lower the probability to guess a matching text for a given hash
  - Key and hash sizes that are considered secure change over time
- **Crypto Cryptography is not security, it is only low probability!? - But "Low" is "VERY LOW"!**
- **Examples of sizes:**

MegaMillions (March, 2012) @ $640Million	**1 In 176,000,000**
DES: 256 different keys	**1 in 72,057,594,037,927,936**
AES-128: 2^{128} different keys	**38 digits**
number of atoms in the earth	**about $1.33*10^{50}$**
number of atoms in our solar system	**about $1.2*10^{57}$**
AES-192: 2^{192} different keys	**58 digits**
AES-256/SHA-256: 2^{256} different keys/hashes	**77 digits**
number of atoms in the universe	**about 10^{82}**
SHA-512: 2^{512} different hashes	**154 digits**

- **Key lengths for symmetric keys are not comparable to those of asymmetric keys**
- **E.g. only a "few" numbers out of 2^{1024} 1024-bit numbers are valid keys for RSA**

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

Hash Example

- **SHA1("The quick brown fox jumps over the lazy dog")**

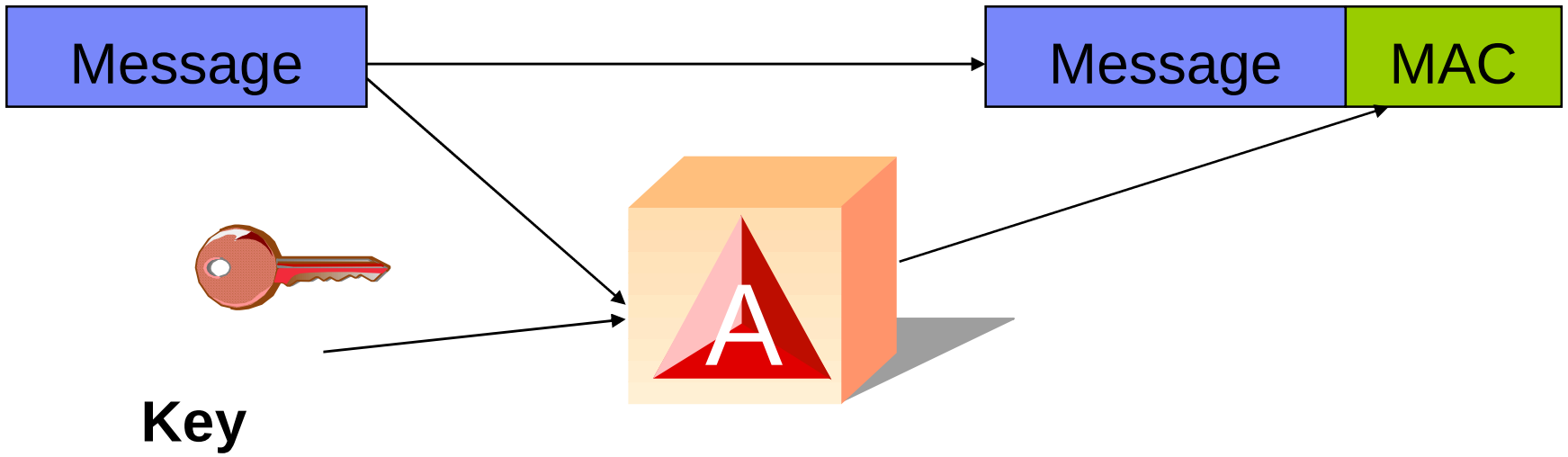
  = A9F1D770 4CA50AC1 504EA60B 12C7AE2D B054AD18

- **SHA1("The quick brown fox jumps over the lazy hog")**

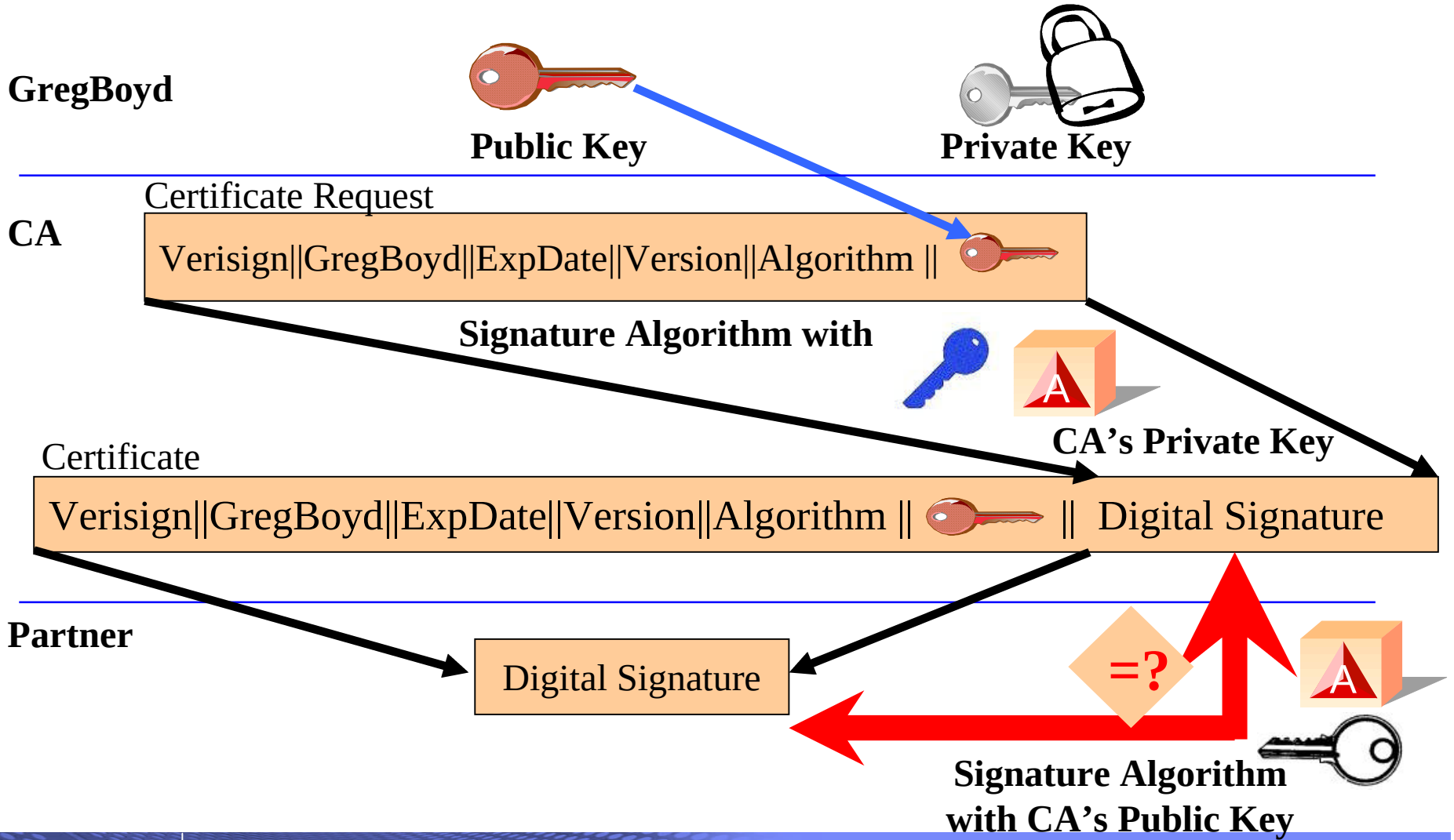
  =24120AD2 C337DE18 24E18BF0 C54F0B4F 98592343

Data Integrity – Message Authentication

Did the message come from who I think it came from?

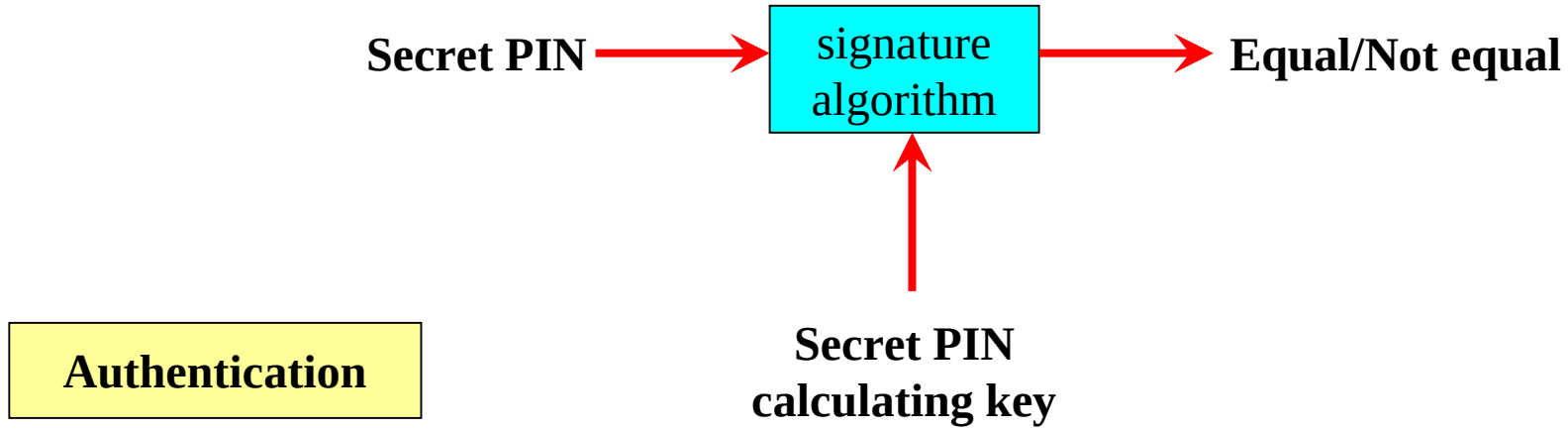


Data Integrity – Digital Certificates



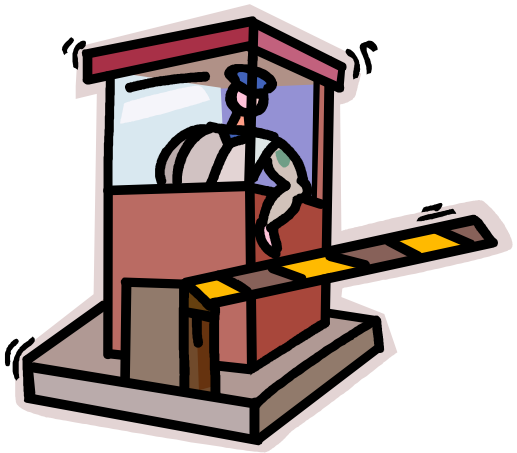
Financial Services

- **PIN Generation**
- **PIN Verification**
- **PIN Export/Import**



Suite B

- **Symmetric Encryption**
  - AES w/key sizes of 128 and 256
- **Digital Signatures**
  - ECDSA – Elliptic Curve, Digital Signature Algorithm
- **Key Agreement**
  - ECDH – Elliptic Curve, 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**

Protected Key – How it works

- **Create a key, with the value ‘ABCD’ and store it as a secure key in the CKDS (i.e. encrypted under the Master Key, MK)**
  - E_{MK}(x’ABCD’) => x’4A!2’ written to the CKDS and stored with a label of MYKEY
- **Execute CSNBSYE (the clear key API to encrypt data), but pass it the key label of our secure key, MYKEY; and text to be encrypted of ‘MY MSG ’**
  - CALL CSNBSYE(….,

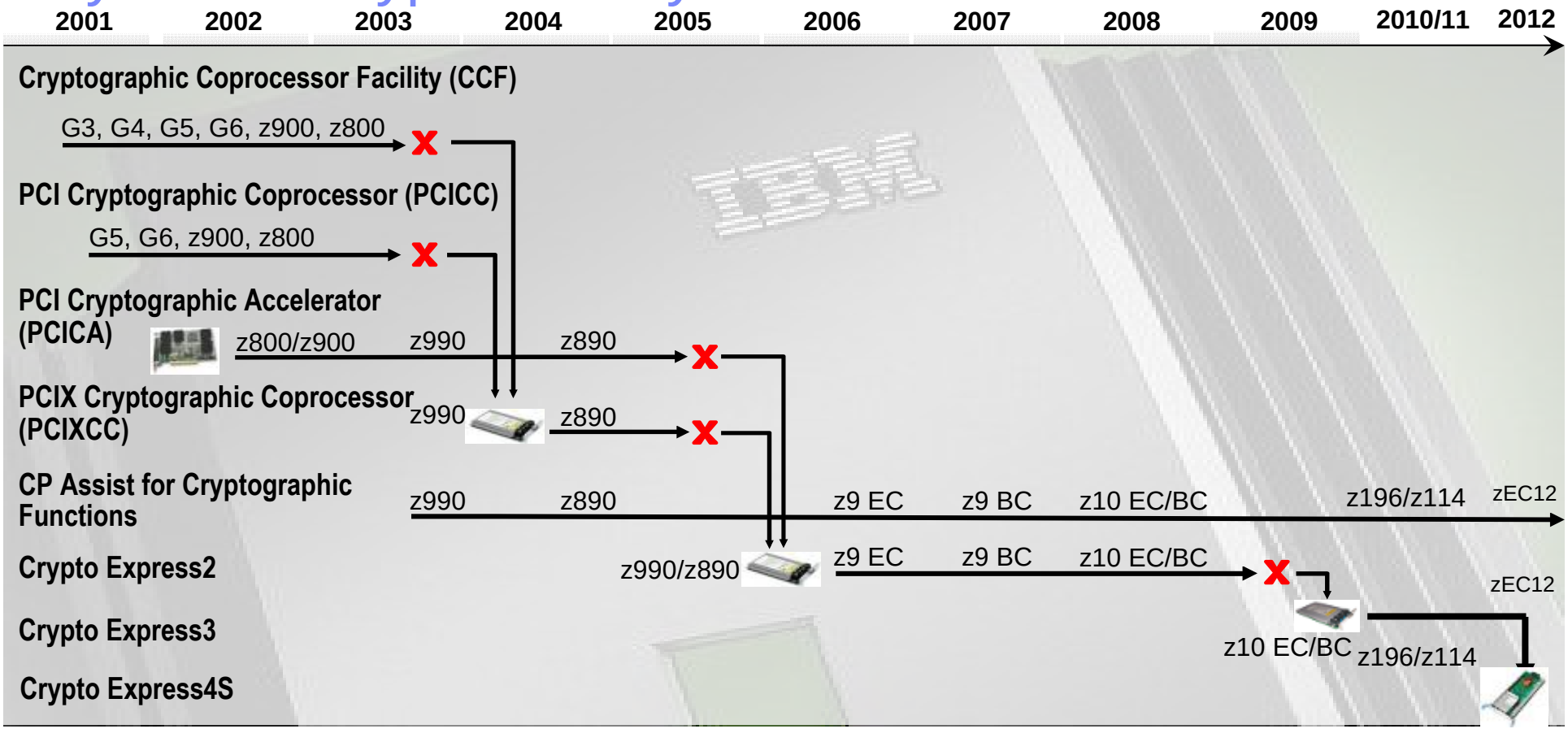
    MYKEY,

    ‘MY MSG ’ ….)

Protected Key – How it works (cont …)

- **ICSF will read MYKEY from the CKDS and pass the key value x'4A!2' to the CEX3**
- **Inside the CEX3, recover the original key value and then wrap it using the wrapping key**
  - D_{MK}(x' 4A!2') => x' ABCD'
  - E_{WK}(x'ABCD') => x'*94E'
- **ICSF will pass the wrapped key value of x'*94E' to the CPACF, along with the message to be encrypted**
- **In the CPACF, we'll retrieve the wrapping key, WK**
  - D_{wk}(x'*94E') => x'ABCD'
  - $E_{x'ABCD'}$('MY MSG ') => ciphertext of x'81FF18019717D183'

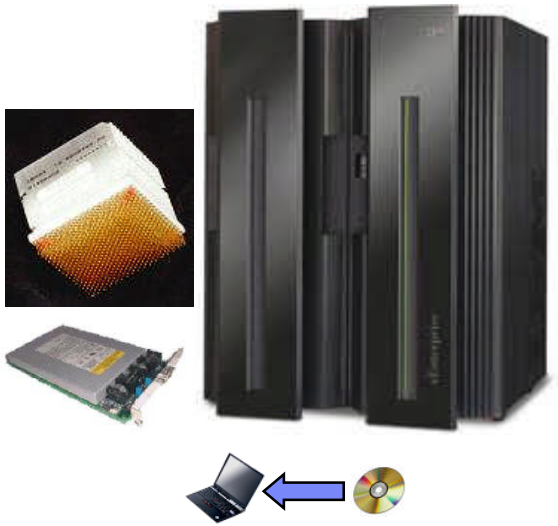
System z Crypto History



- 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 – PCIe Interface, additional processing capacity with improved RAS
- Crypto Express4S - IBM Standard PKCS #EP11

System z Clear Key Crypto Hardware – z9 (EC & BC), *z10 (EC (GA3) & BC (GA2))*, *z196/z114*, *zEC12*

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



- **Single CPACF per CP on zEC12 & z9**

TechDoc WP100810 – A Synopsis of System z Crypto Hardware

System z Secure Key Crypto Hardware

- **Secure Key DES/TDES (z9)**
- **Secure Key AES (z9)**
- **Financial (PIN) Functions (z9)**
- **Key Generate/Key Management (z9)**
- **Random Number Generate and Generate Long (z9)**
- Protected Key Support (CEX3 on z10)
- **SSL Handshakes (z9), ECDSA support (CEX3 on z196)**
- EP11 (CEX4S)

TechDoc WP100810 – A Synopsis of System z Crypto Hardware

Master Keys

- **ICSF uses five master keys to protect operational keys**
  - DES Master Key (DES-MK aka SYM-MK)

    - 128 bit key / 192 bit key
    - Protects DES/TDES (symmetric) application keys
  - AES Master Key (AES-MK)

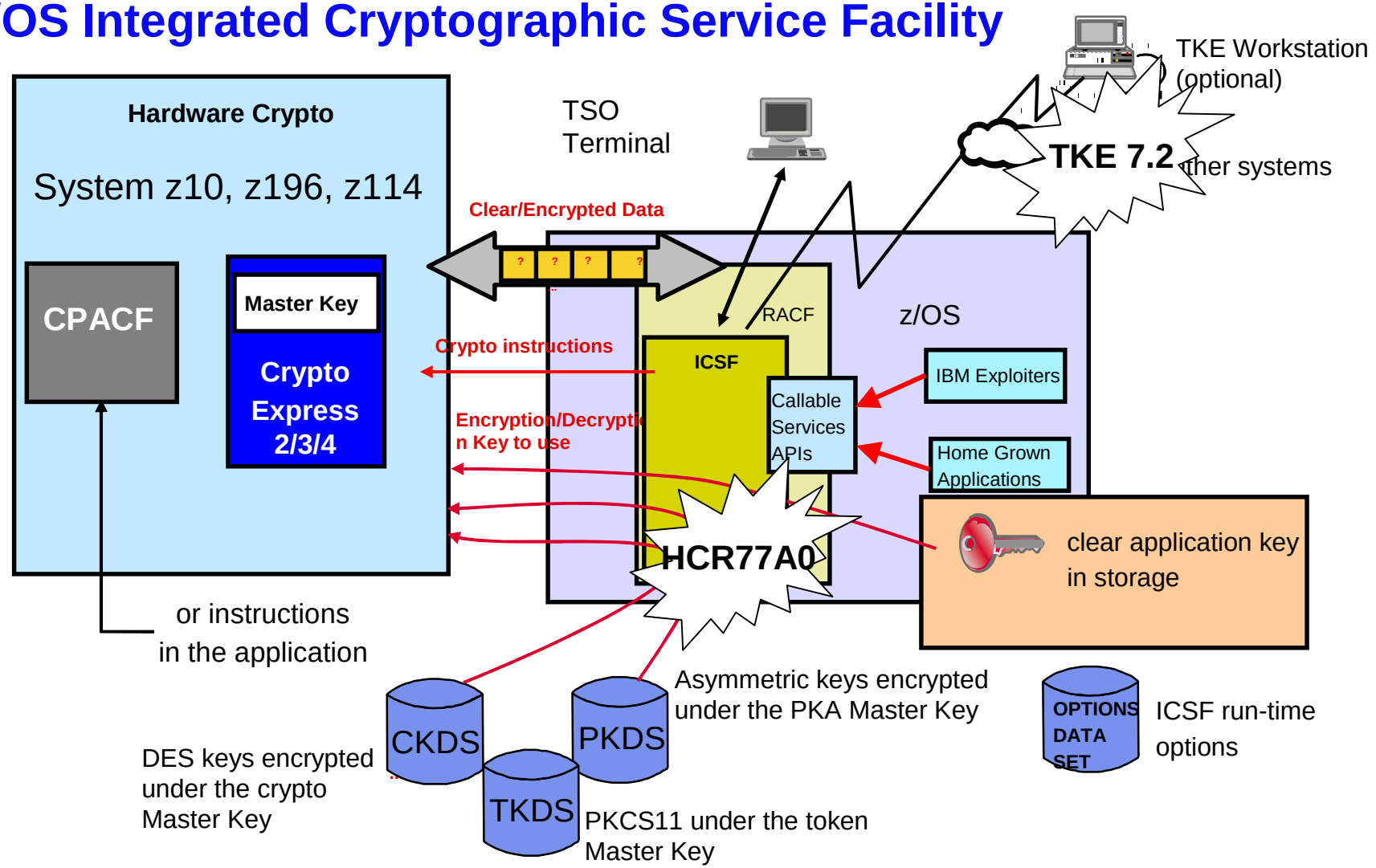
    - 256 bit key
    - Protects AES (symmetric) application keys
  - Asymmetric-keys master key (RSA-MK aka ASYM-MK)

    - 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
- **Stored within the secure hardware boundary of the cryptographic coprocessor**

z/OS Integrated Cryptographic Service Facility

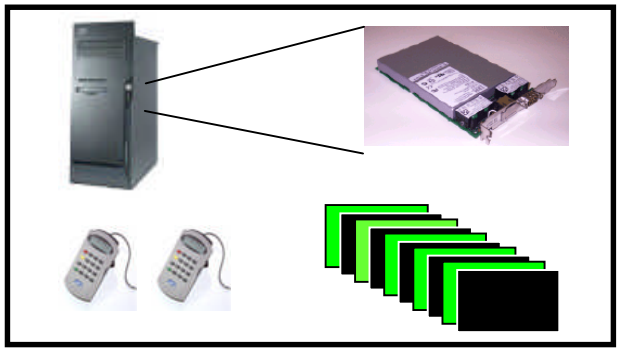


Access to the cryptographic services and keys can be controlled by RACF with the CSFSERV and CSFKEYS classes

Trusted Key Entry (TKE) Workstation

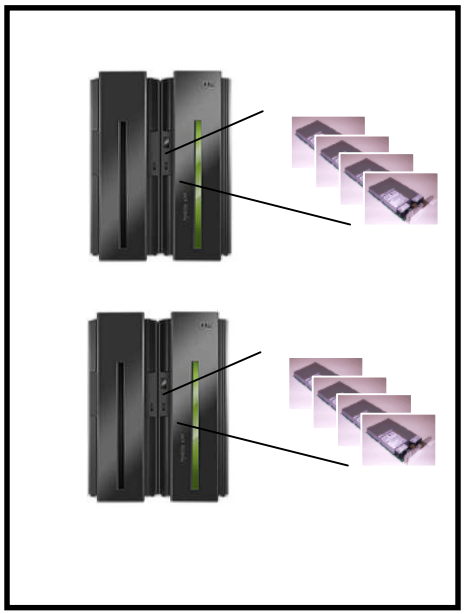
Components

- 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 (zEC12, z196, z114 and z10 EC and BC) from a single point of control
  - Support requirements for standards
  - Simplification of tasks



Enhancements

- Support of new hardware or firmware functions
  - Support for Crypto Express4S defined as a CCA coprocessor
  - Required for Crypto Express4S defined as an Enterprise PKCS #11coprocessor
  - New DES operational keys
  - New AES CIPHER key attribute
  - Allow creation of corresponding keys
  - Support 4 smart card readers
- Support requirements for standards
  - Stronger key wrapping

Linux on System z Crypto Stack

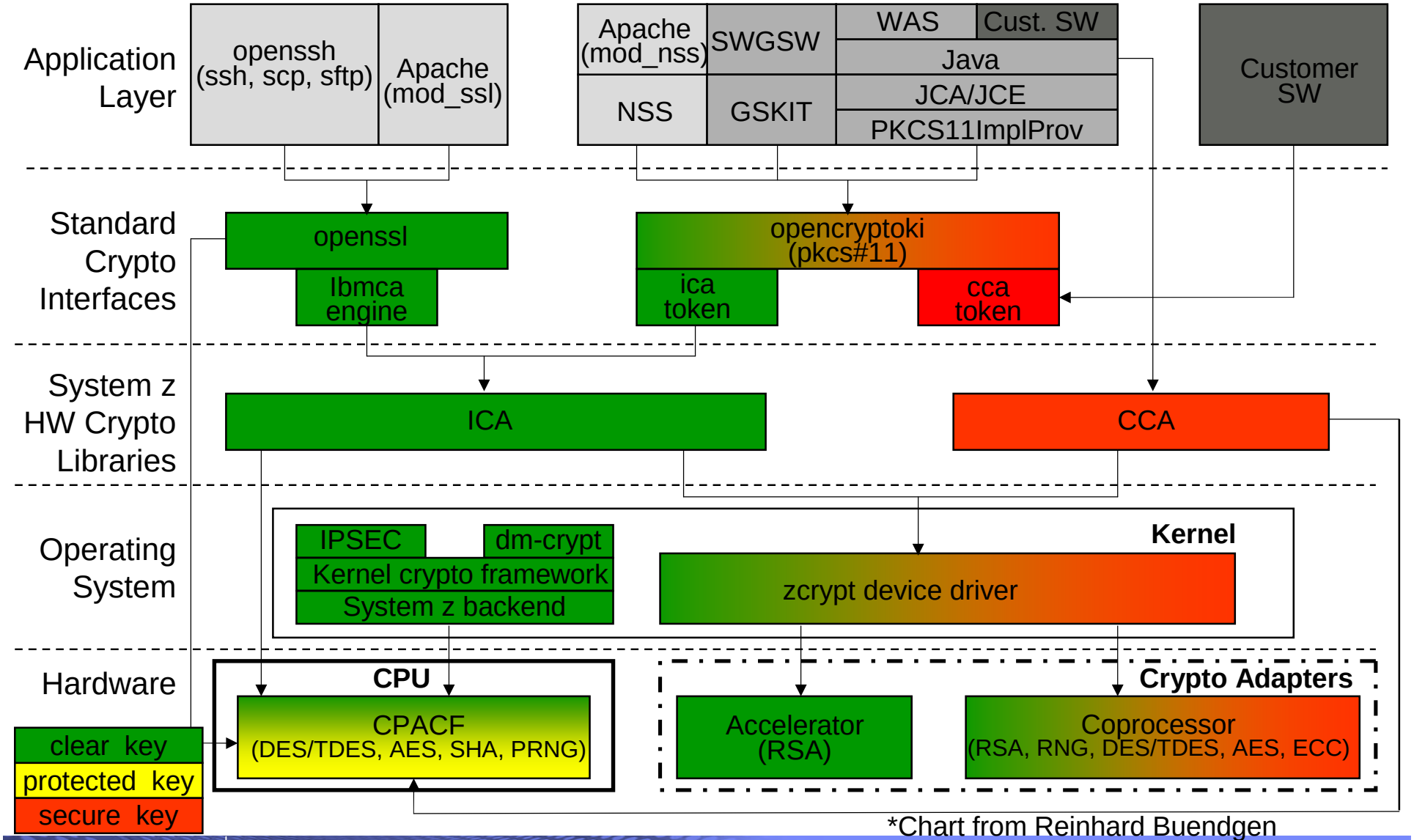
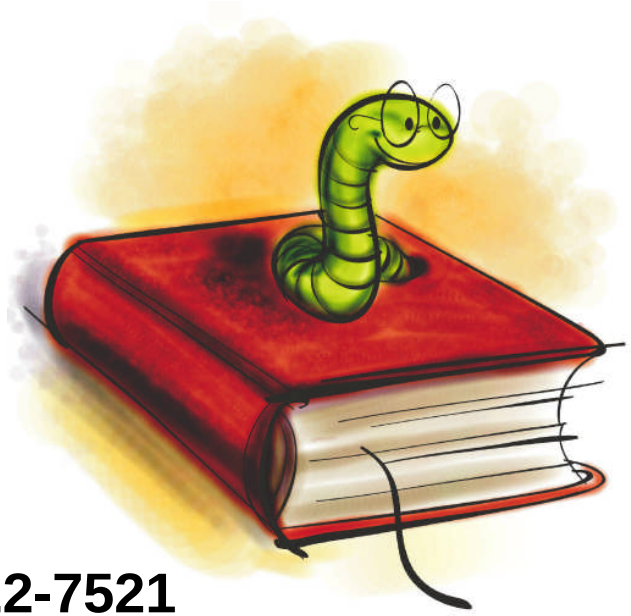


*Chart from Reinhard Buendgen

References

- **Cryptography Books**
  - Bruce Schneier, ‘Applied Cryptography Second Edition: Protocols, Algorithms, and Source Code in “C”’, Addison Wesley 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
  - www.csrc.nist.gov – Computer Security Resource Center of NIST
  - www.rsasecurity.com/rsalabs - Research site for RSA Security
- **Free Stuff**
  - www.scmagazine.com - SC Magazine
  - www.counterpane.com – Bruce Schneier web site with monthly newsletter

IBM Pubs



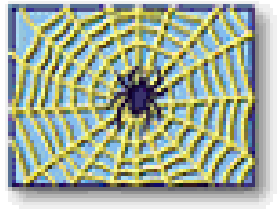
- **ICSF Overview, SA22-7519**
- **ICSF Administrator's Guide, SA22-7521**
- **ICSF Application Programmer's Guide, SA22-7522**
- **ICSF System Programmer's Guide, SA22-7520**

z/OS Web Download Site

http://www.ibm.com/systems/z/os/zos/downloads/

IBM Resources (on the web)

- **Redbooks – www.redbooks.ibm.com 'Crypto'**
  - IBM zEnterprise EC12 Technical Introduction, SG24-8050
  - IBM zEnterprise EC12 Technical Guide, SG24-8049
  - IBM zEnterprise 196 Configuration Setup, SG24-7834
  - IBM zEnterprise System Technical Introduction, SG24-7832
  - IBM zEnterprise 196 Technical Guide, SG24-7833
  - IBM zEnteprise 114 Technical Guide, SG24-7954
  - System z Crypto and TKE Update, SG24-7848



- **ATS TechDocs Web Site www.ibm.com/support/techdocs (Search All Documents for keyword of 'Crypto')**
  - WP100810 – A Synopsis of System z Crypto Hardware
  - WP100647 – A Clear Key/Secure Key/Protected Key Primer

Linux Secure Key Crypto - Information & Download

- Crypto Cards Hardware Overview - http://www.ibm.com/security/cryptocards/
- PCI-E Cryptographic Coprocessor (CEX3) - http://www.ibm.com/security/cryptocards/pciecc/overview.shtml
- Crypto Product Summary – http://www.ibm.com/security/cryptocards/pcixcc/overproduct.shtml
- CCA Overview - http://www.ibm.com/security/cryptocards/pcixcc/overcca.shtml
- PCI-E Cryptograhpic Coprocessor Library - http://www.ibm.com/security/cryptocards/pciecc/library.shtml
- CCA Library Download – http://www.ibm.com/security/cryptocards/pciecc/ordersoftware.shtml
- Cryptographic Coprocessor Order Information - http://www.ibm.com/security/cryptocards/pciecc/order.shtml
- PCI-E Cryptographic Coprocessor Support - http://www.ibm.com/security/cryptocards/pciecc/support.shtml
- Cutting Edge Cryptography by Peter Spera, May/June 2007 IBM Systems Magazine - http://www.ibmsystemsmag.com/mainframe/mayjune07/features/14233p1.aspx

Questions …

Trademarks

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

AlphaBlox*	GDPS*	RACF*	Tivoli*
APPN*	HiperSockets	Redbooks*	Tivoli Storage Manager
CICS*	HyperSwap	Resource Link	TotalStorage*
CICS/VSE*	IBM*	RETAIN*	VSE/ESA
Cool Blue	IBM eServer	REXX	VTAM*
DB2*	IBM logo*	RMF	WebSphere*
DFSMS	IMS	S/390*	zEnterprise
DFSMShsm	Language Environment*	Scalable Architecture for Financial Reporting	xSeries*
DFSMSrmm	Lotus*	Sysplex Timer*	z9*
DirMaint	Large System Performance Reference™ (LSPR™)	Systems Director Active Energy Manager	z10
DRDA*	Multiprise*	System/370	z10 BC
DS6000	MVS	System p*	z10 EC
DS8000	OMEGAMON*	System Storage	z/Architecture*
ECKD	Parallel Sysplex*	System x*	z/OS*
ESCON*	Performance Toolkit for VM	System z	z/VM*
FICON*	PowerPC*	System z9*	z/VSE
FlashCopy*	PR/SM	System z10	zSeries*

* Registered trademarks of IBM Corporation

Processor Resource/Systems Manager

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.
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.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.
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.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
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.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective 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.

QR Code