Hans-Joachim Picht, Linux on System z Initiatives & Linux Champion EMEA <hans@de.ibm.com>

Crypto Support for Linux on System z - Introduction

WAVV Conference
Colorado Springs, Colorado, April 15-19, 2011 - Crowne Plaza, Colorado Springs

Agenda

- Why Cryptography
- System z cryptographic hardware
  - CP Assist for Cryptographic Function (CPACF)
  - System z Crypto cards
- Accelerated Linux kernel functions
- Pseudo Random Number Generator
- File system encryption: eCryptfs
- lszcrypt & chzcrypt
- Cryptographic Libraries
- Apache SSL Setup
- Java
- Summary

The Importance of Security

Loss of customer data at BNY Mellon much bigger than first thought

Bank confirms tape with info on 12 million customers of its shareholder service unit is unaccounted for

BNY MELLON

Sept 2, 2008

Massive insider breach at DuPont

A research chemist who worked for DuPont for 10 years before accepting a job with a competitor downloaded 22,000 sensitive documents

DUPONT®

Feb 15, 2007

Societe Générale loses $7.2 billion in trading fraud

Lack of privileged password management and insufficient IT security controls

Jan 24, 2008

SOCIETE GENERALE

Why cryptography?

Regulation & Compliance

- PCI – Payment Card Industry
- Standards for all organizations
- Debit Cards and Credit Cards
- Sarbanes-Oxley (SOX), etc.

Secure messaging and Non-repudiation

- Needs certificate management system.
- Public Key Infrastructure (PKI).

Money management

- ATMs, PINs, Chip & PIN cards.
- Banking and eCommerce on the internet
- Secure signon.

Off-site backups

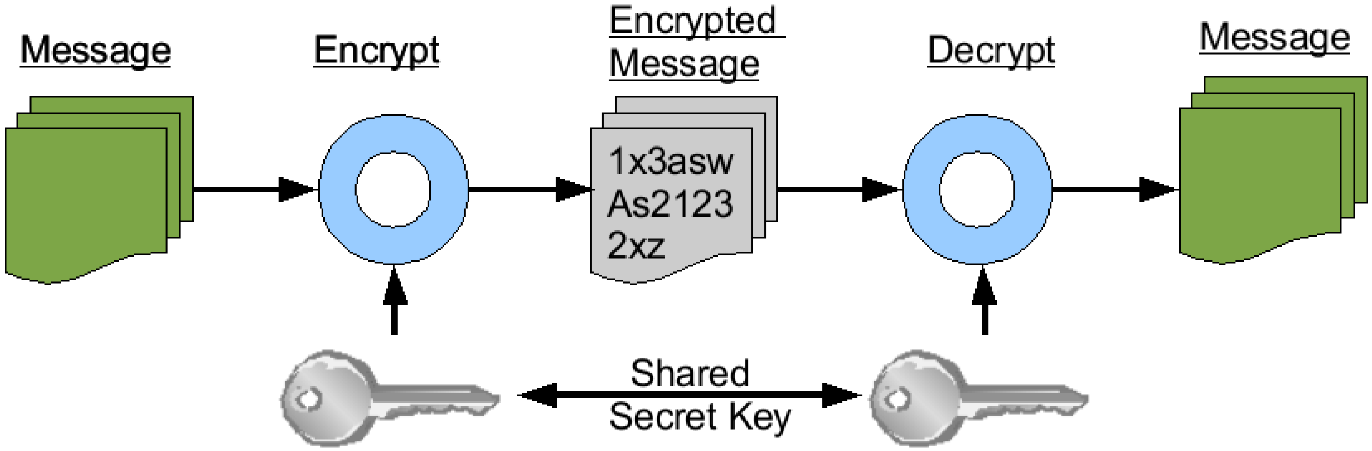
- Do tapes fall off your lorries?

....just to name a few reasons...

What types of cryptography are there?

Symmetric Algorithms

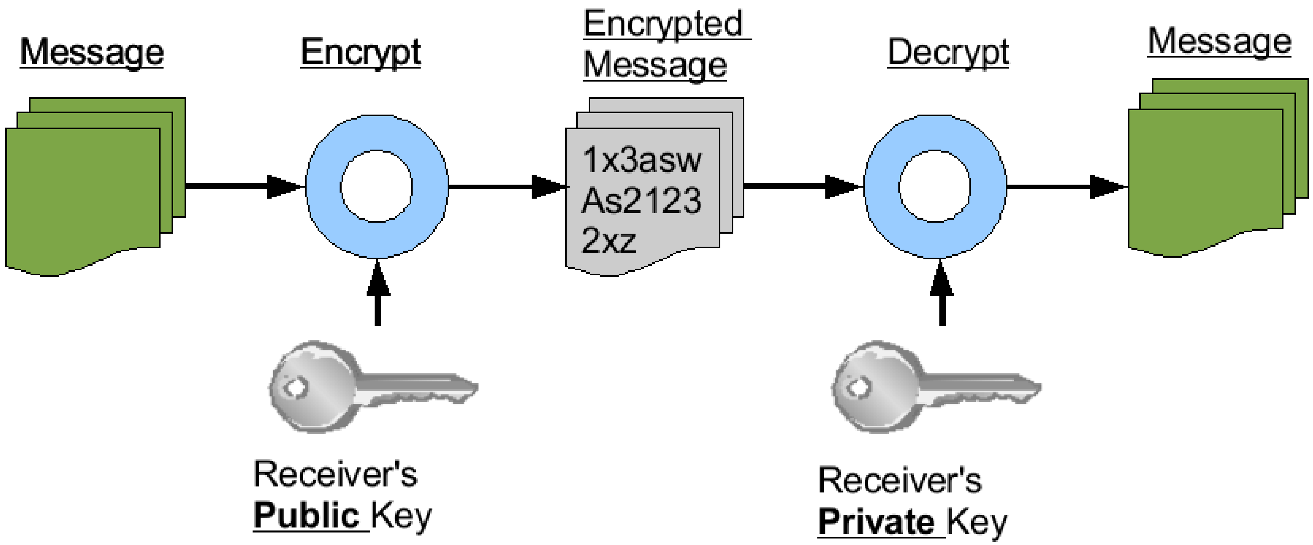
- Same key for encrypt and decrypt
- Relatively fast algorithms
- Used to encrypt the data
- Problem is key exchange
- Examples: DES, 3-DES, AES, IDEA, Twofish



What types of cryptography are there? (cont.)

Asymmetric Algorithms

- Key-pair, one key for encrypt and one for decrypt (public & private key)
- Relatively slow algorithms
- Used for key exchange and digital signatures
- Examples: RSA, ElGamal, Elliptic curve cryptosystem, Diffie-Hellman

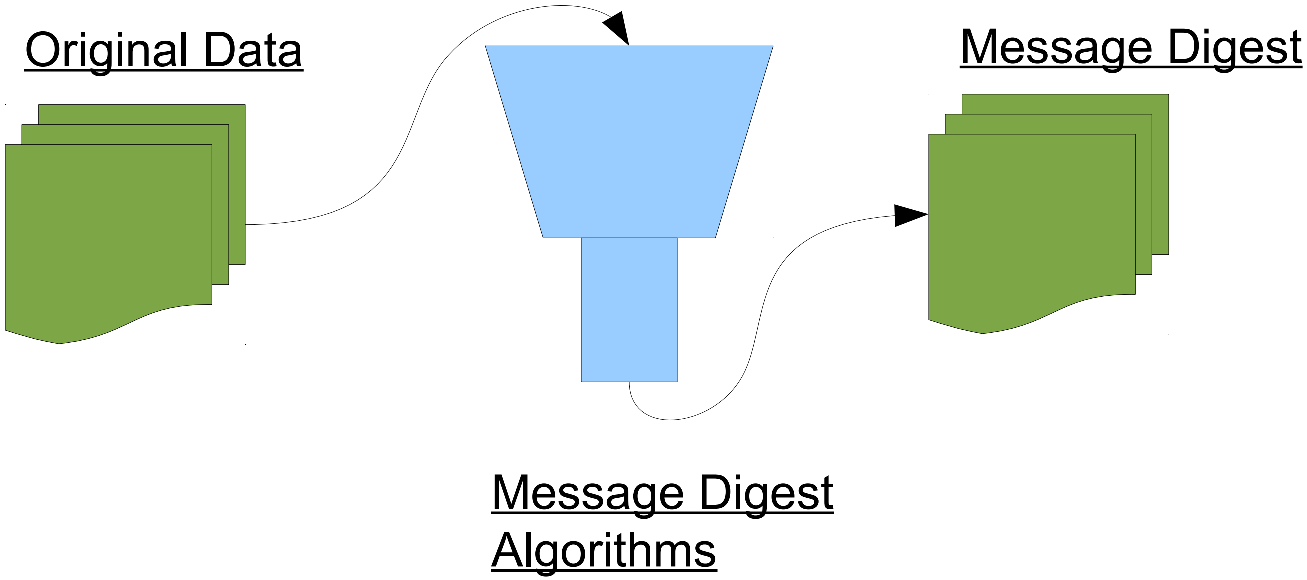


Note: Usually Asymmetric cryptography is much slower that symmetric cryptography

What types of cryptography are there? (cont.)

Message Digest / Hashing Algorithms

- Used for message integrity (message authentication)
- Examples: MD5, SHA-1, SHA-512



Original Data

Message Digest

Message Digest Algorithms

System z cryptographic hardware

System z has two flavours for accelerating cryptographic operations:

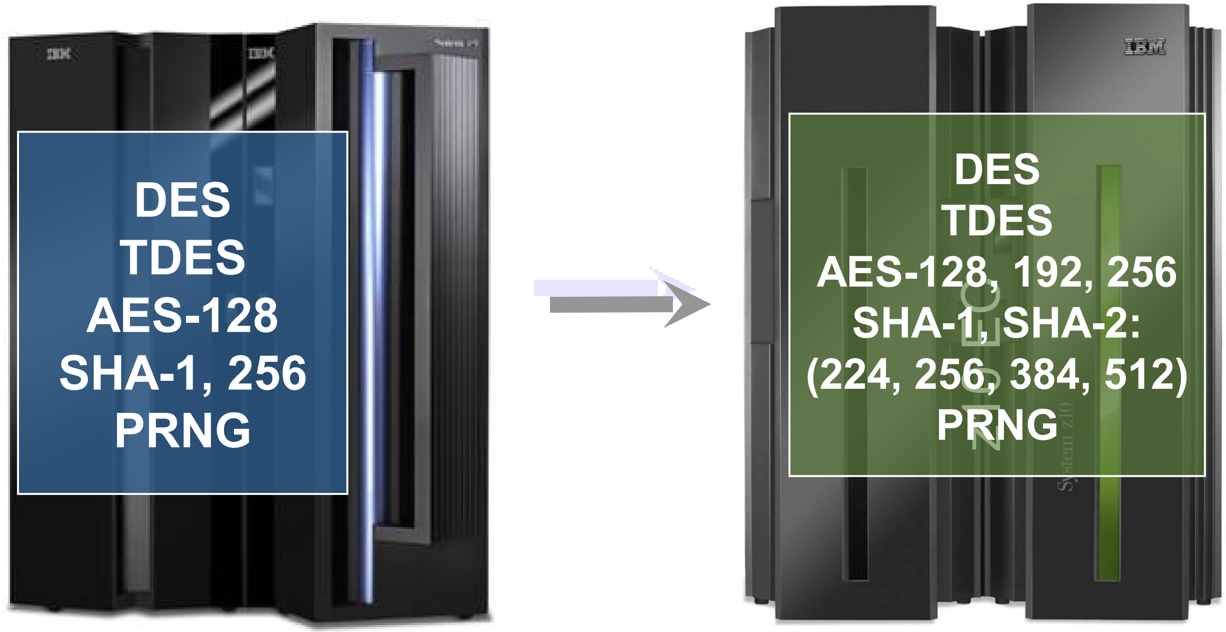
- CP assists for symmetric algorithms (CPACF)
- Crypto cards for asymmetric algorithms

Purpose:

- move cryptographic workload away from central processor
- accelerate encryption / decryption
- achieve higher security level

CP Assist for Cryptographic Function (CPACF)

- CPACF is a no-charge enablement feature on System z hardware.
- CPU support for symmetric algorithms is included in every standard CP & IPL



DES
TDES
AES-128
SHA-1, 256
PRNG

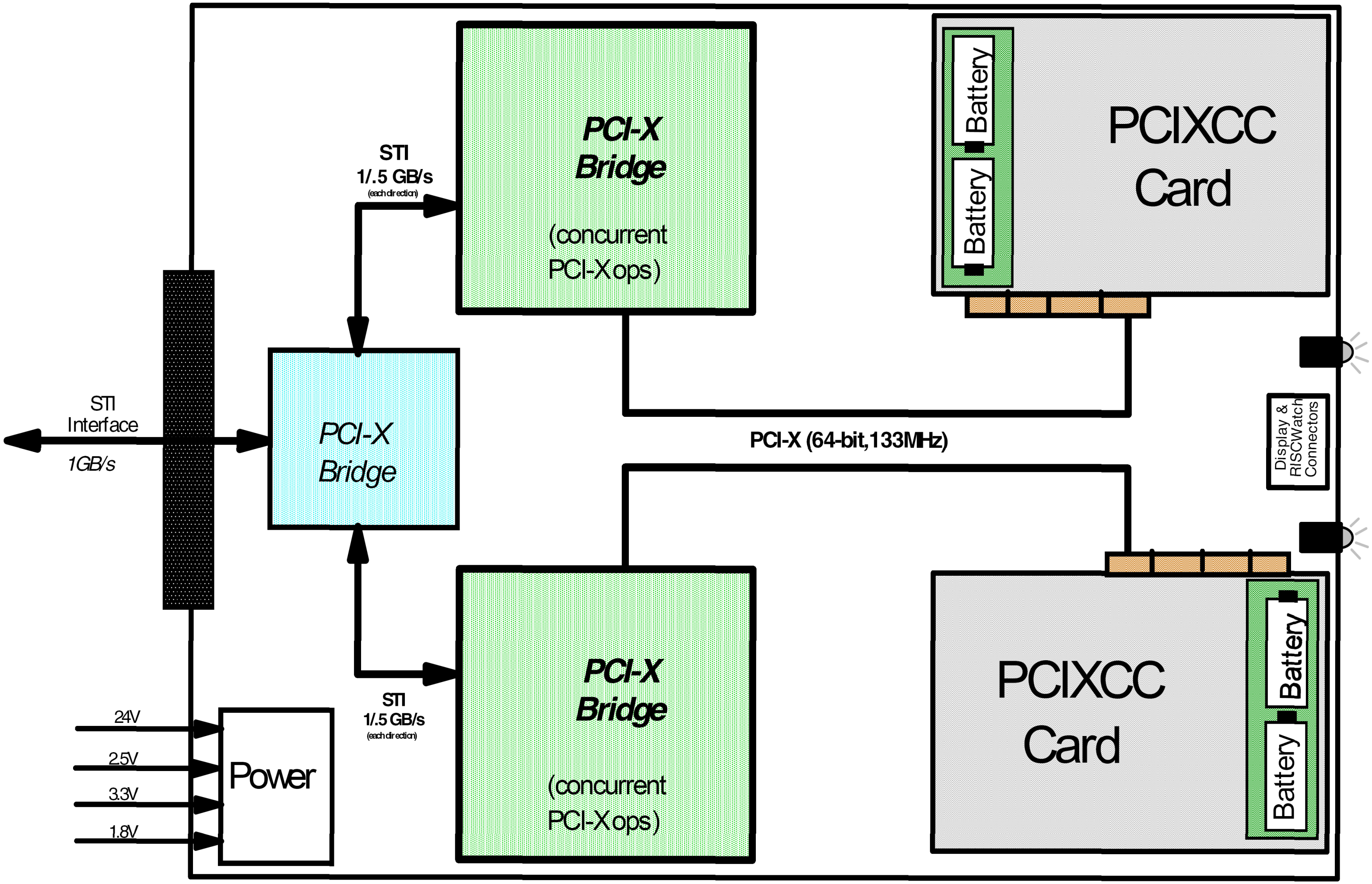
DES
TDES
AES-128, 192, 256
SHA-1, SHA-2:
(224, 256, 384, 512)
PRNG

If you plan to enable CPACF, consider that it can only be enabled or disabled on a per central processor complex (CPC) basis and will affect all logical partitions (LPAR). The CPACF feature code is 3863. For information about how to enable a feature code see System z10 Support Element Operations Guide , SC28-6979.

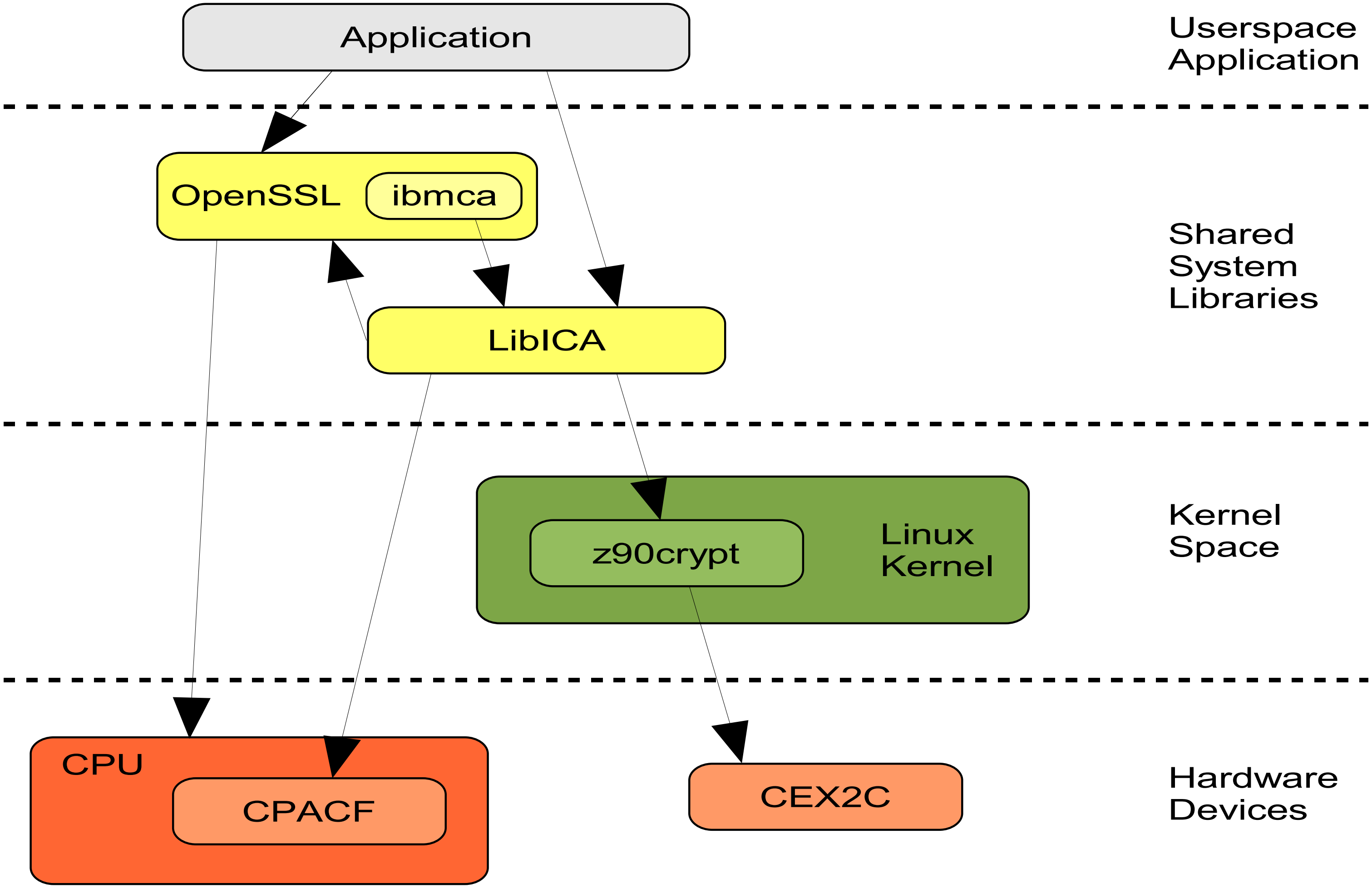
System z Crypto cards

- Coprocessor vs. accelerator
  - Coprocessor: secure key transactions
  - Accelerator: SSL acceleration
- PCI Cryptographic Coprocessor (PCICC)
- PCI Cryptographic Accelerator (PCICA)
- PCI-X Cryptographic Coprocessor (PCIXCC)
- Crypto Express2 Coprocessor (CEX2C)
- Crypto Express2 Accelerator (CEX2A)
- z9-109: PCIXCC & PCICA -> Crypto Express 2 (CEX2)
  - 2x PCI-X adapter, configurable as coprocessor or accelerator
- If you are running Linux under z/VM, version 5.1 or later is required
- For the newer cards (CEX2A and CEX2C) you require a System z9 or later

System z Crypto cards (cont.)



Linux on System z Cryptography Support Overview



Distribution Support

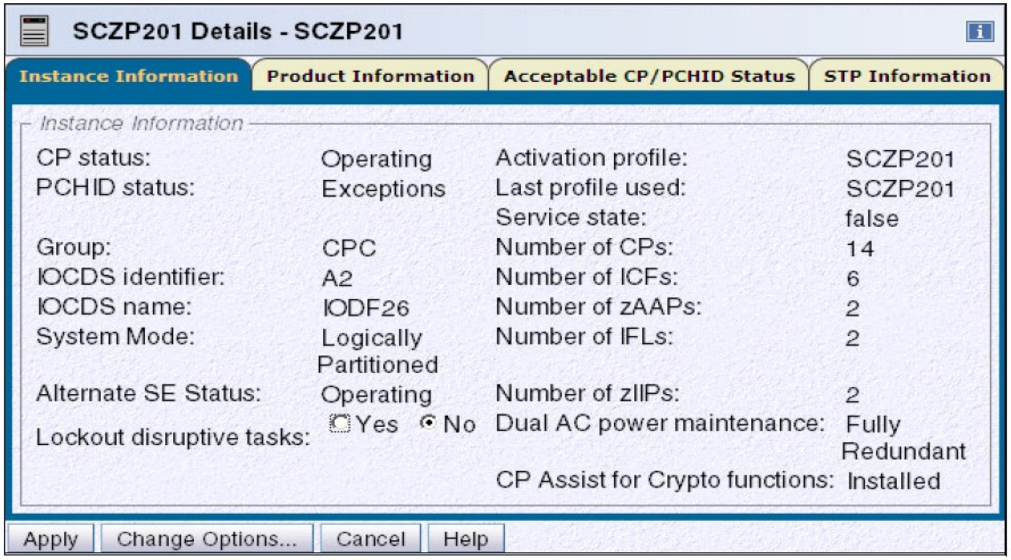
- **SLES and RHEL ship with libica, openCryptoki and openssl-ibmca support**
- **Red Hat Enterprise Linux (RHEL)**
  - RHEL 4: z90crypt, CPACF with z990 / z890 support, CEX2A support with U4
  - RHEL 5: z9 CPACF included (SHA1 not usable), PRNG in U1, new zcrypt driver with U1, z10 CPACF with U2
- **Novell SUSE Enterprise Linux Server (SLES)**
  - SLES9 : z90crypt driver, CPACF with z990 / z890 support
  - SLES10 : new zcrypt driver with SP1, z9 CPACF included, PRNG in SP1 z10 CPACF with SP2
  - SLES11 : new zcrypt driver & z9/z10 CPACF

Enabling CPACF

To check whether CPACF is enabled for your CPC, open the CPC's details in the Support Element interface and check the value of "CP Assist for Crypto functions" in the lower right corner of the dialog.

- If the value indicates "Installed" as in screenshot, CPACF is enabled.
- After CPACF has been enabled, you can use the `icainfo` tool from the libica package on Linux to verify that your Linux guest has access to the hardware-supported cryptography functions.

```
root@larsson:~>  icainfo
The following CP Assist for
Cryptographic Function
(CPACF) operations are
supported by libica on this
system:
SHA-1: yes
SHA-256: yes
SHA-512: yes
DES: yes
[...]
```



SCZP201 Details - SCZP201

Instance Information

CP status: Operating
PCHID status: Exceptions
Group: CPC
IOCDS identifier: A2
IOCDS name: IODF26
System Mode: Logically Partitioned
Alternate SE Status: Operating
Lockout disruptive tasks: ○ Yes ● No

Activation profile: SCZP201
Last profile used: SCZP201
Service state: false
Number of CPs: 14
Number of ICFs: 6
Number of zAAPs: 2
Number of IFLs: 2
Number of zIIPs: 2
Dual AC power maintenance: Fully Redundant
CP Assist for Crypto functions: Installed

Crypto Express support for z/VM user

- To check, from CP, whether your LPAR definition includes both a CEX2C and a CEX2A configuration, issue a QUERY CRYPTO AP statement.

```
#CP QUERY CRYPTO AP
AP 00 CEX2A Queue 11 is installed
AP 07 CEX2C Queue 11 is dedicated to LNXRH2
```

- Access from Linux to the Crypto Express adapter is provided by the `z90crypt` Linux kernel module.
- Use `lsmod` command to determine whether the `z90crypt` module is loaded. If not, use `modprobe` to load it and check the `/dev` file system to verify that the z90crypt device was created successfully

```
root@larsson:~> lsmod | grep z90crypt || echo “not loaded”
not loaded
root@larsson:~> modprobe z90crypt
root@larsson:~> lsmod | grep z90crypt || echo “not loaded”
z90crypt 106668 0
root@larsson:~> ls -l /dev/z90crypt
crw-rw-rw- 1 root root 10, 61 Feb 14 23:44 /dev/z90crypt
```

Troubleshooting

- If `modprobe` fails, you probably have not defined in the user directory and Linux has no access to any Crypto Express adapter

```
root@larsson:~> modprobe z90crypt
FATAL: Error inserting z90crypt
(/lib/modules/[...]/s390/crypto/z90crypt.ko): No such device
root@larsson:~> dmesg | tail -1
The hardware system does not support AP instructions
```

- To verify that a virtual cryptographic device has been defined for your z/VM user, issue the QUERY VIRTUAL CRYPTO through the `vmcp` command

```
root@larsson:~> vmcp QUERY VIRTUAL CRYPTO
AP 47 CEX2A Queue 06 shared
```

- Successfully loading the z90crypt driver creates a new file in the procfs directory under `/proc/driver/z90crypt` which provides statistical information about the driver and the underlying (virtual) hardware.

```
[root@larsson ~]# cat /proc/driver/z90crypt
[...]
PCICA count: 1
PCICC count: 0
[...]
```

Accelerated Linux kernel functions

- The standard Linux kernel includes modules that exploit the CPACF capabilities of the System z hardware.
- This enhances the generic Linux crypto support
  - generic, platform independent implementation of several symmetric encryption algorithms (see next slide)
  - support for optimized implementations
  - used by kernel functions, e.g. IPsec, dm-crypt
- The hardware-dependent modules for the CPACF are typically located in the `/lib/modules` directory unless they have been compiled into the kernel itself

```
[root@larsson ~]# ls -l /lib/modules/`uname -r`/kernel/arch/s390/crypto/
total 292
drwxr-xr-x 2 root root 4096 Feb 11 12:10 ./
drwxr-xr-x 6 root root 4096 Feb 11 12:10 ../
-rwxr--r-- 1 root root 36664 Feb 18 16:11 aes_s390.ko*
-rwxr--r-- 1 root root 27392 Feb 18 16:11 des_check_key.ko*
[...]
```

Accelerated Linux kernel functions (cont)

```
[root@larsson ~]# modprobe
aes_generic
[root@larsson ~]# modprobe aes_s390
[root@larsson ~]# cat /proc/crypto
name          : cbc(aes)
driver        : cbc-aes-s390
module        : aes_s390
priority      : 400
refcnt        : 1
type          : blkcipher
blocksize     : 16
min keysize   : 16
max keysize   : 16
ivsize        : 16
name          : ecb(aes)
driver        : ecb-aes-s390
module        : aes_s390
priority      : 400
refcnt        : 1
[...]
```

Symmetric algorithms and hash Functions

CPACF currently includes:

- SHA-1, DES and 3-DES with z990 / z890
- SHA-256 and AES with 128 with z9
- SHA-512 and AES with 192/256 bit keys
- with z10

Algorithms exploiting CP assist functions can be used instead of generic software implementation

AES fall-back support for unsupported key lengths on z9

Pseudo random number generator

Algorithm description at /proc/crypto

Pseudo Random Number Generator

Another important aspect of cryptography is the availability of enough entropy to ensure secrecy because many cryptographic functions rely on randomly chosen values that are used, for example, to generate session keys or initialize the internal pseudo random number generator (PRNG).

- The PRNG driver is a character device that provides user-space applications with pseudo-random numbers generated by the pseudo-random number generator of the CPACF (based on the 3-DES CPACF algorithm)
- PRNG provides pseudo-random numbers similar to the Linux pseudo-random number device `/dev/urandom` but provides a better performance

The module has two parameter:

- prng_chunk_size: The number of bytes that the Linux kernel reads from the hardware in a single read operation (independent from the number of bytes a user-space program reads
- entropy_limit: The number of bytes read from the hardware after which extra randomness is added to the state of the pseudo random number generator

```
[root@larsson ~]# modprobe prng
[root@larsson ~]# modprobe prng_chunk_size=256
[root@larsson ~]# modprobe prng_entropy_limit=4096
```

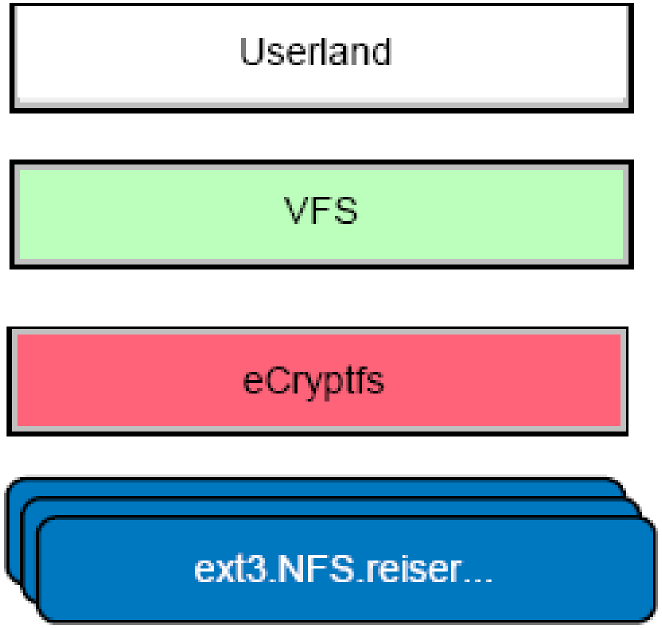
Pseudo random number generator (cont.)

- You can test the speed of the new pseudorandom number generator by using the `dd` tool.
- The following shows that the CPACF-assisted random number generator is roughly twice as fast as the default generator that is being used for the `/dev/urandom` device.

```
[root@larsson ~]# dd if=/dev/random of=/dev/null bs=4k
0+3 records in
0+3 records out
24 bytes (24 B) copied, 11.0261 seconds, 0.0 kB/s
[root@larsson ~]# dd if=/dev/urandom of=/dev/null bs=4k
14975+0 records in
14974+0 records out
61333504 bytes (61 MB) copied, 11.0188 seconds, 5.6 MB/s
[root@larsson ~]# dd if=/dev/prandom of=/dev/null bs=4k
34746+0 records in
34745+0 records out
142315520 bytes (142 MB) copied, 11.0129 seconds, 12.9 MB/s
```

File system encryption: eCryptfs

- The additional cryptographic capabilities of System z can also be used to speed up the process of protecting your file system data using encryption.
- Native to the kernel, eCryptfs is a stacked cryptographic file system for Linux.
  - A stacked file system is layered on top of an existing mounted file system, which is referred to as a lower file system.
  - As the files are written to or read from the lower file system, eCryptfs encrypts and decrypts the files.
  - eCryptfs automatically asks you a few questions about the encryption properties when you set up the file system.
  - The CPACF supports the aes, des, and des3_ede128 modes.
  - All other encryption is performed in software only!



File system encryption: eCryptfs (cont.)

```
[root@larsson ~]# mkdir /mnt/{un,}encrypted
[root@larsson ~]# mount -t ecryptfs /mnt/encrypted /mnt/unencrypted
Select key type to use for newly created files:
1) openssl
2) tspi
3) passphrase
Selection: 3
Passphrase: ****
Select cipher:
1) aes: blocksize = 16; min keysize = 16; max keysize = 32 (loaded)
[...]
Selection [aes]: aes
Select key bytes:
1) 16
2) 32
3) 24
Selection [16]: 24
[...]
```

Additional Solutions

Dm-crypt

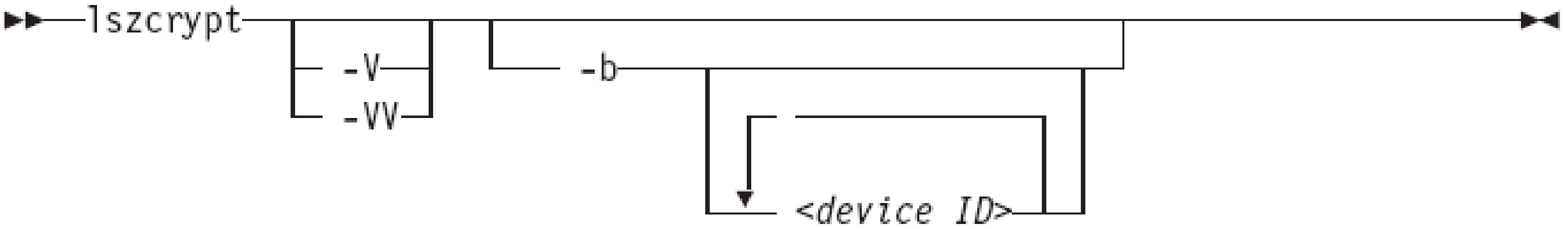
- One of the other cryptography features of the Linux kernel is dm-crypt, which is a device mapper and a transparent disk encryption subsystem that allows you to encrypt whole block devices.
- For more information about dm-crypt, see: http://www.saout.de/tikiwiki/tiki-index.php

IPSec

- The IPSec protocol is used to encrypt on-wire IP traffic to counter eavesdropping and tampering by third parties.
- IPSec uses standard kernel cryptography and benefits from the System z hardware acceleration with CPACF.
- For more information about running IPSec with Linux, see: http://www.ipsec-howto.org

lszcrypt

Use the **lszcrypt** command to display information about cryptographic adapters managed by zcrypt and zcrypt's AP bus attributes



- To display card type and online status of all available cryptographic adapters:

```
root@larsson:~> lszcrypt -V
```

- To display card type, online status, hardware card type, hardware queue depth, and request count for cryptographic adapters 0, 1, 10, and 12

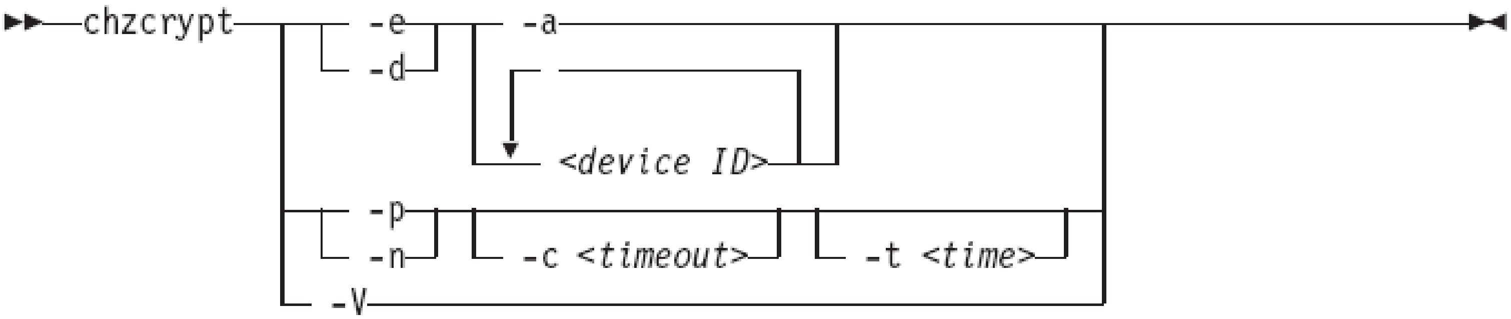
```
root@larsson:~>  lszcrypt -VV 0 1 10 12
```

- To display AP bus information:

```
root@larsson:~> lszcrypt -b
```

chzcrypt

Use the **chzcrypt** command to configure cryptographic adapters managed by zcrypt and modify zcrypt’s AP bus attributes.



- To set the cryptographic adapters 0, 1, 4, 5, and 12 online:

```
root@larsson:~>  chzcrypt -e 0 1 4 5 12
```

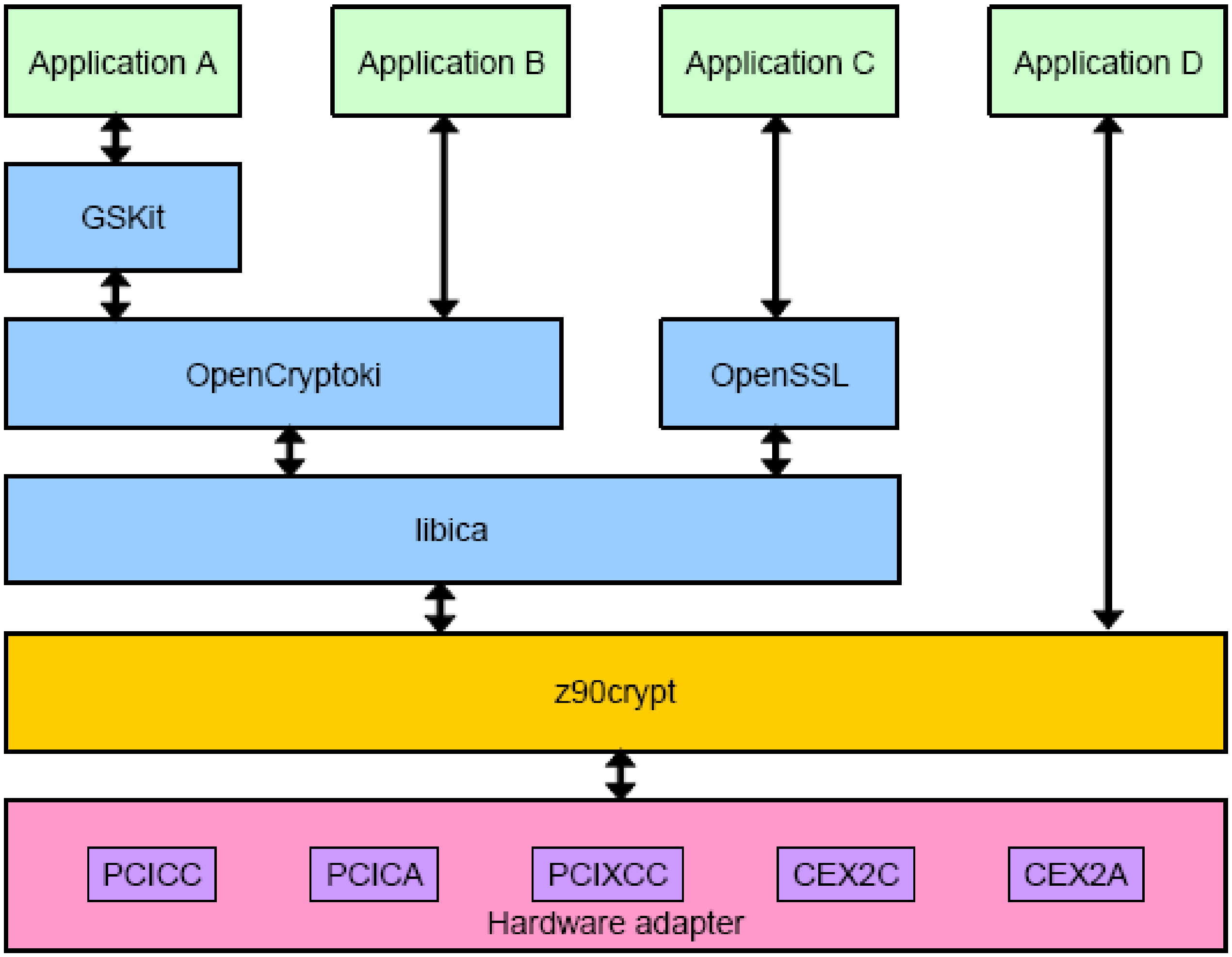
- To set all available cryptographic adapters offline:

```
root@larsson:~>  chzcrypt -d -a
```

- To set the configuration timer for re-scanning the AP bus to 60

```
root@larsson:~>  chzcrypt -c 60 -n
```

Cryptographic Libraries



Key Management

One important aspect of cryptography is managing the keys that give access to the unencrypted data.

The Crypto Express2 and Crypto Express3 adapters in accelerator mode can be used as an Hardware Security Module (HSM) that protects your private keys from being read even by legitimate key users.

- The only difference to a full featured HSM, like the Crypto Express card in coprocessor mode, is that the encrypted keys are still stored in the file system of the user operating system.
- This situation implies that you take the necessary precautions to prevent the key from being deleted or overwritten, and implement a backup process according to your companies policies.
- If you need a full HSM, you can use the Crypto Express adapters in coprocessor mode.
- The most widespread standard for access tokens to security modules is PKCS#11 from the RSA Laboratories, and is nicknamed Cryptoki.

Cryptographic Libraries: openCryptoki

openCryptoki is an open source package which provides a implementation of the PKCS #11 API that allows interfacing to devices (such as a smart card, smart disk, or PCMCIA card) that hold cryptographic information and perform cryptographic Functions.

- openCryptoki provides application portability by isolating the application from the details of the cryptographic device.
- Both Red Hat Enterprise Linux 5.4 and SUSE Linux Enterprise Server 11 include the necessary tools to initialize and access the security tokens.
- OpenCryptoki provides exploitation of cryptographic hardware via libICA

Before you start working with the PKCS#11 API, be sure that the z90crypt module is loaded and the /dev/z90crypt device is present.

- The next step is to set a new PIN on the slot that you want to use, initialize the token, and set a user PIN on the token - using the `pkcsconf` application

More Information:

- http://sourceforge.net/projects/opencryptoki/
- http://www.ibm.com/developerworks/linux/library/s-pkcs/

Network Security Services (NSS) library

- Starting with version 5.4, Red Hat Enterprise Linux supports Network Security Services (NSS), a library for managing cryptographic objects and performing cryptographic functions in a platform independent way.

- Applications built with NSS can support SSL v2 and v3, TLS, PKCS #5, PKCS #7, PKCS #11, PKCS #12, S/MIME, X.509 v3 certificates, and other security standards.

- NSS started out as a Netscape project and is now being developed by the Mozilla Foundation.

- It is also used for key management by other vendors, for example in the Sun Solaris operating system.

- More Information: http://www.mozilla.org/projects/security/pki/nss/

Cryptographic Libraries: GSkit

To access and manage your PKCS#11 tokens, you can also use the Global Security Toolkit (GSKit) from IBM, an API that provides platform-independent functions for secure communication using SSL.

- GSKit supports access to cryptographic tokens and functions since version 7. and is bundled with other IBM software
  - IBM HTTP-Server
  - Tivoli Access Manager
  - Tivoli Directory Server
  - Websphere MQ

One of the utilities that is provided by GSKit is iKeyman.

- It allows you to effectively manage key stores through the GSKit interfaces and supports various key store formats by using a plug-in mechanism.
- One of the plug-ins enables GSKit to access PKCS#11 key stores, and you can use it to access and manipulate the contents of the security tokens managed by openCryptoki.

See the respective manuals of those software packages for a description of how to install GSKit.

Cryptographic Libraries: OpenSSL

OpenSSL is most likely the most prominent open source (licensed under BSD-style license) cryptographic library

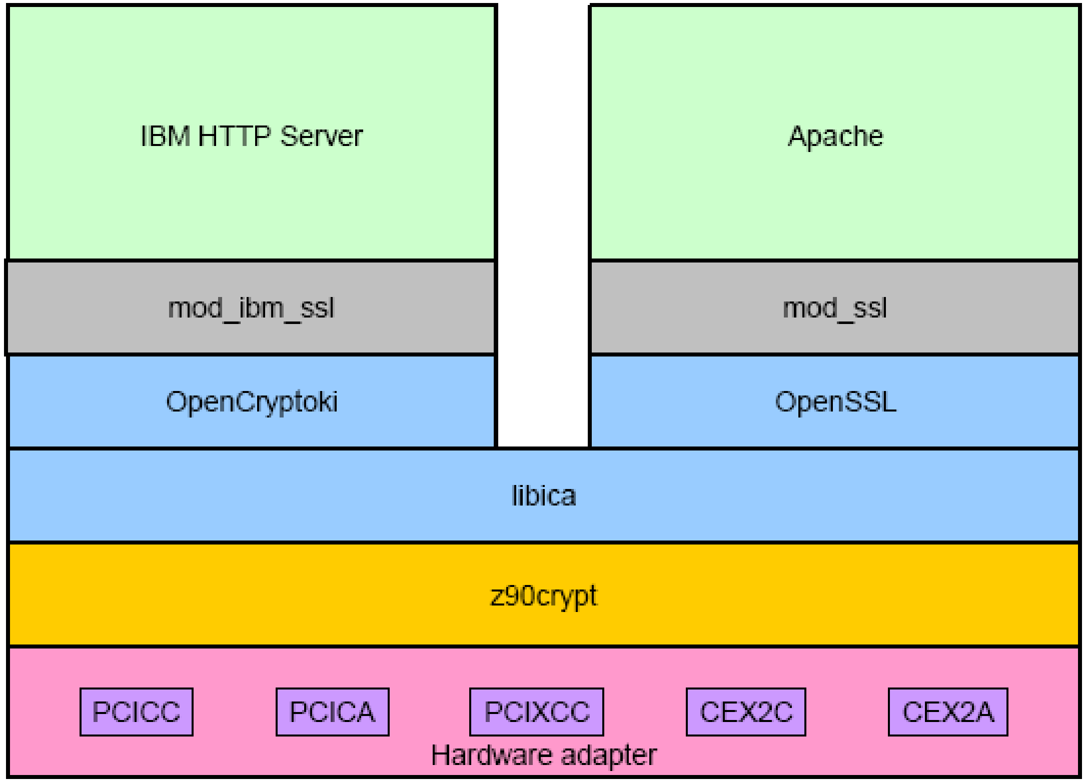
The core library implements the basic cryptographic functions and provides various utility functions:

- The OpenSSL library has a plug-in mechanism that allows various engines to be used for cryptographic operations.
- One of these engines is the ibmca engine
- It is usually located in: `/usr/lib/openssl/engines/libibmca.so` and uses CPACF and Crypto Express functionality if they are present in the system
- More Information: http://www.openssl.org/

```
root@larsson:~> openssl engine -c
(dynamic) Dynamic engine loading support
(ibmca) Ibmca hardware engine support
 [RSA, DSA, DH, RAND, DES-ECB, DES-CBC, DES-EDE3, DES-EDE3-
  CBC, AES-128-ECB, AES-128-CBC, AES-192-ECB, AES-192-CBC,
  AES-256-ECB, AES-256-CBC, SHA1]
```

Apache-SSL Setup

- The Apache HTTP Server, commonly referred to simply as Apache is a popular open source web server AND supports a variety of features, including SSL support
- We can utilize hardware cryptographic devices to improve performance during SSL handshake negotiation



RHEL: To enable SSL with Apache 2.0:

- Load the z90crypt device driver & ensure that OpenSSL is installed and enabled
- Add a certificate to the key database.
- The `/etc/httpd/conf/httpd.conf` file contains two directives to configure the key database:
  - The `SSLCertificateFile` directive specifies the SSL certificate.
  - The `SSLCertificatekeyFile` directive specifies the key file for the SSL Certificate.
- By default, these point to a fake self-signed certificate and key
- suitable tor testing.
- OpenSSL provides an interface to create a new certificate and key.
- Create a new certificate and key files in the `/etc/httpd/conf` directory

RHEL: Apache-SSL Setup (cont.)

```
root@larsson:~> cd /etc/httpd/conf
root@larsson:~> mkdir ssl.csr
root@larsson:~> mkdir ssl.key; mkdir ssl.crt
root@larsson:~> openssl req -new > ./ssl.csr/ap2.cert.csr
root@larsson:~> cd /etc/httpd/conf; mkdir ssl.csr
root@larsson:~> mkdir ssl.key; mkdir ssl.crt
root@larsson:~> openssl req -new > ./ssl.csr/ap2.cert.csr
root@larsson:~> openssl rsa -in privkey.pem -out \
./ssl.key/ap2.cert.key
root@larsson:~> openssl x509 -in ./ssl.csr/ap2.cert.csr \
-out ./ssl.crt/ap2.cert.cert -req \
-signkey ./ssl.key/ap2.cert.key -days 365
```

RHEL: Apache-SSL Setup (cont.)

- Remove the passphrase from the key (saving the key in a key file):

```
root@larsson:~> openssl rsa -in privkey.pem -out
./ssl.key/ap2.cert.key
```

- Convert the request to a signed certificate:

```
root@larsson:~> openssl x509 -in ./ssl.csr/ap2.cert.csr
-out \      ./ssl.crt/ap2.cert.cert -req \
-signkey ./ssl.key/ap2.cert.key -days 365
```

- Point to the new certificate and key in the /etc/httpd/conf/httpd.conf:

```
SSLCertificateFile /etc/httpd/conf/ssl.crt/ap2.cert.cert
SSLCertificatekeyFile /etc/httpd/conf/ssl.key/ap2.cert.key
```

- SSL configuration directives are contained in the /etc/httpd/conf.d/ssl.conf file. The “SSLCryptoDevice” directive defines which engine interface is enabled:

```
SSLCryptoDevice ibmca
```

- Activate the webserver

```
root@larsson:~> apachectl start
```

SLES: To enable SSL with Apache 2.0:

- Load the z90crypt device driver
- Edit the /etc/sysconfig/apache2 file
  - The APACHE_MODULES=”module_list” directive must include “ssl” in module_list.
  - The APACHE_SERVER_FLAGS=”SSL” directive must include “SSL”.
- Run SuSEConfig
- Generate a server certificate or import an existing server certificate (e.g. /usr/share/packages/apache2/certificates.sh)
- Create a virtual host configuration file
- Copy a template virtual host configuration file to the virtual host configuration directory:
  - # cp /etc/apache2/vhost.d/vhost-ssl.template /etc/apache2/vhost.d/vhost ssl.conf
  - Edit the vhost-ssl.conf file and add the location of the server certificate and the serverprivate key.

SLES: Apache-SSL Setup (cont.)

Configure Apache 2.0 to use the ibmca OpenSSL engine interface.

Edit the `/etc/apache2/ssl-global.conf` file and set the SSLCryptoDevice directive to

```
<IfDefine SSL>
<IfDefine !NOSSL>
<IfModule mod_ssl.c>
SSLCryptoDevice ibmca
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl
```

- The server can be started and stopped as a service using the rcapache command (rcapache start)

Apache & PRNG

The hardware-supported PRNG can be used in any application that uses `/dev/random` or `/dev/urandom` to read random values. One example is the Apache httpd server program where you can use it for seeding the SSL or NSS pseudorandom number generator.

SLES 11

- By default, the Apache server in SUSE uses mod_ssl module for serving content over an encrypted communication channel.
- Adjust mod_ssl to access the PRNG device that utilizes CPACF and modify the file /etc apache2/ssl-global.conf configuration file as follows

```
SSLRandomSeed startup file:/dev/prandom 1024
SSLRandomSeed connect file:/dev/prandom 512
```

RHEL 5.4

- Red Hat Enterprise Linux includes both mod_ssl and mod_nss for encrypted communication.
- Modify the `/etc/httpd/conf.d/nss.conf` configuration as following:

```
NSSRandomSeed startup file:/dev/prandom 512
```

Java

The hardware cryptography extensions provided by both CPACF and the Crypt Express adapters are also available to you in the Java programming environment, enabling you to build and run applications with increased performance.

- Access to the hardware is mediated by the IBM Java PKCS#11 implementation (IBMPKCS11Impl) library that is, for example, provided with the IBM Java Runtime Environment (JRE).

More Information:

http://www.ibm.com/developerworks/java/jdk/security/50/secguides/pkcs11implDocs/IBMJavaPKCS11ImplementationProvider.html

Summary

Security is more than a “Perimeter” defense” - a firewall alone is not suffici

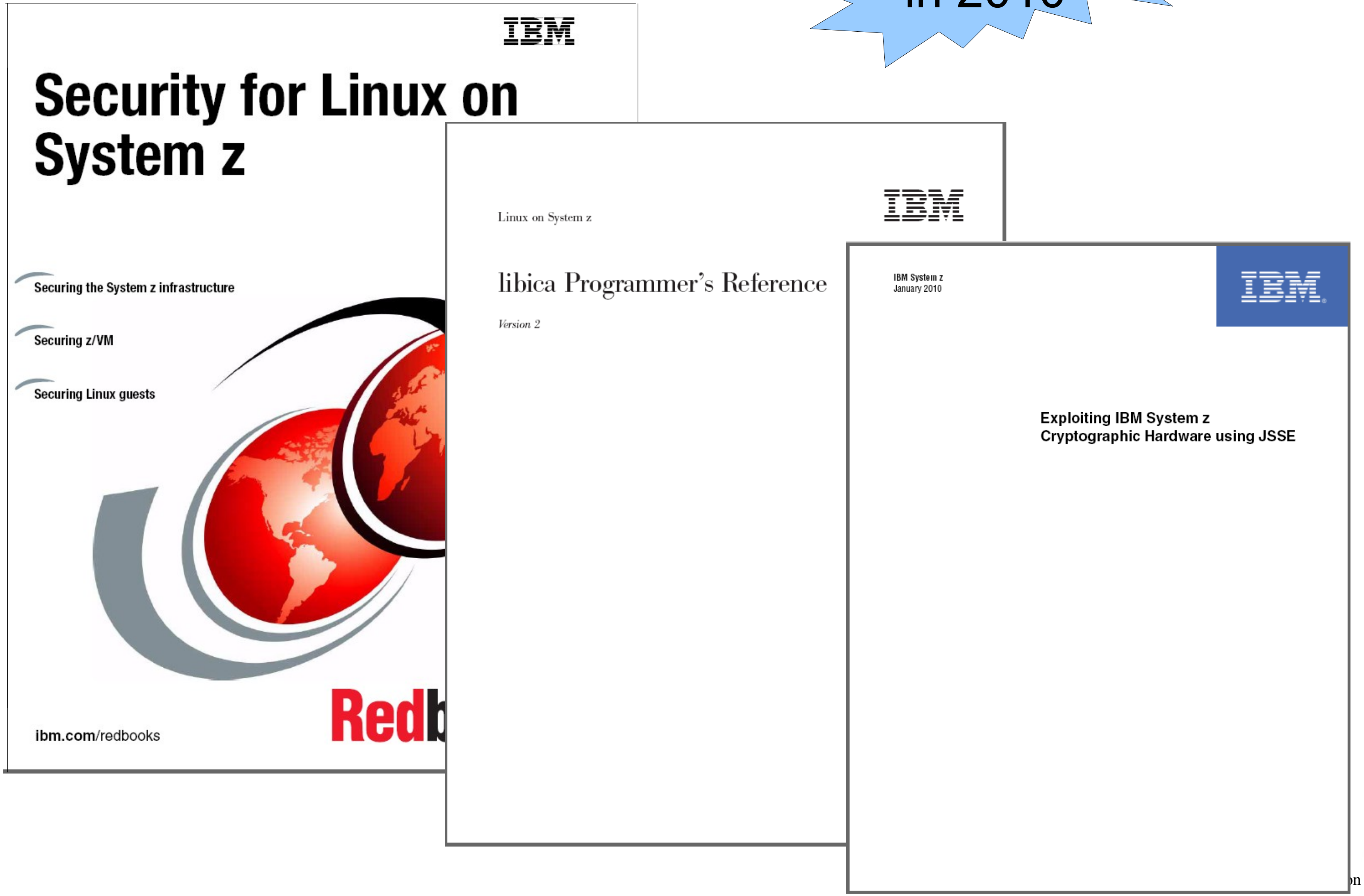
Security begins with the security capabilities / functions available within Enterprise infrastructure

Linux running on System z leverages:

- Unique Hardware Features
- Support for trusted cryptography algorithms
- Secure open source implementation
- A software layer to make use of the functionality from the application layer

More Information

Updated in 2010



Security for Linux on System z

Securing the System z infrastructure

Securing z/VM

Securing Linux guests

ibm.com/redbooks

Redb

Linux on System z

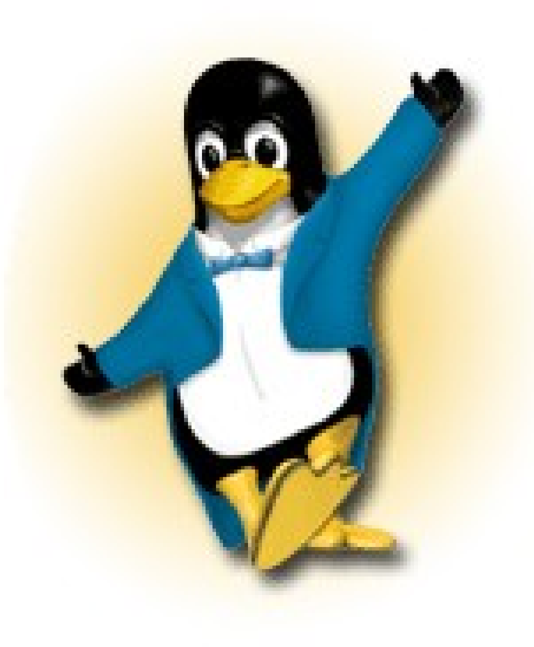
libica Programmer's Reference

Version 2

IBM System z
January 2010

Exploiting IBM System z
Cryptographic Hardware using JSSE

Questions?



Hans-Joachim Picht

Linux on System z Initiatives

IBM Deutschland Research
& Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-1810
Mobile +49 (0)175 - 1629201
hans@de.ibm.com

Trademarks & Disclaimer

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml: AS/400, DB2, e-business logo, ESCON, eServer, FICON, IBM, IBM Logo, iSeries, MVS, OS/390, pSeries, RS/6000, S/390, System Storage, System z9, VM/ESA, VSE/ESA, WebSphere, xSeries, z/OS, zSeries, z/VM.

The following are trademarks or registered trademarks of other companies

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries. LINUX is a registered trademark of Linux Torvalds in the United States and other countries. UNIX is a registered trademark of The Open Group in the United States and other countries. Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation. SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC. Intel is a registered trademark of Intel Corporation. * 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.

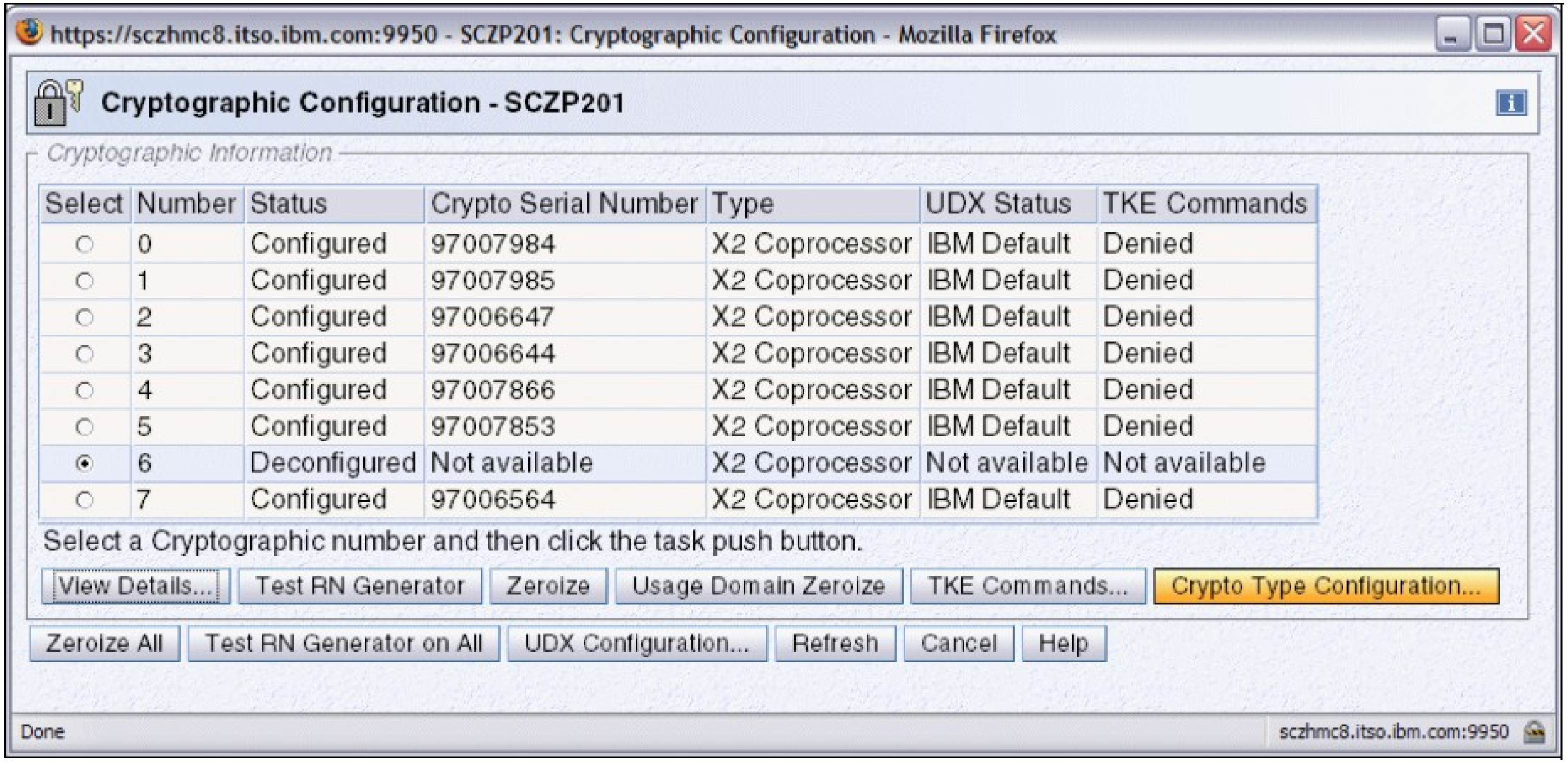
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply. All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they 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. References in this document to IBM products or services do not imply that IBM intends to make them available in every country. Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use. The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice. Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Appendix

LPAR hardware setup: CryptoExpress2 adapter



https://sczhmc8.itso.ibm.com:9950 - SCZP201: Cryptographic Configuration - Mozilla Firefox

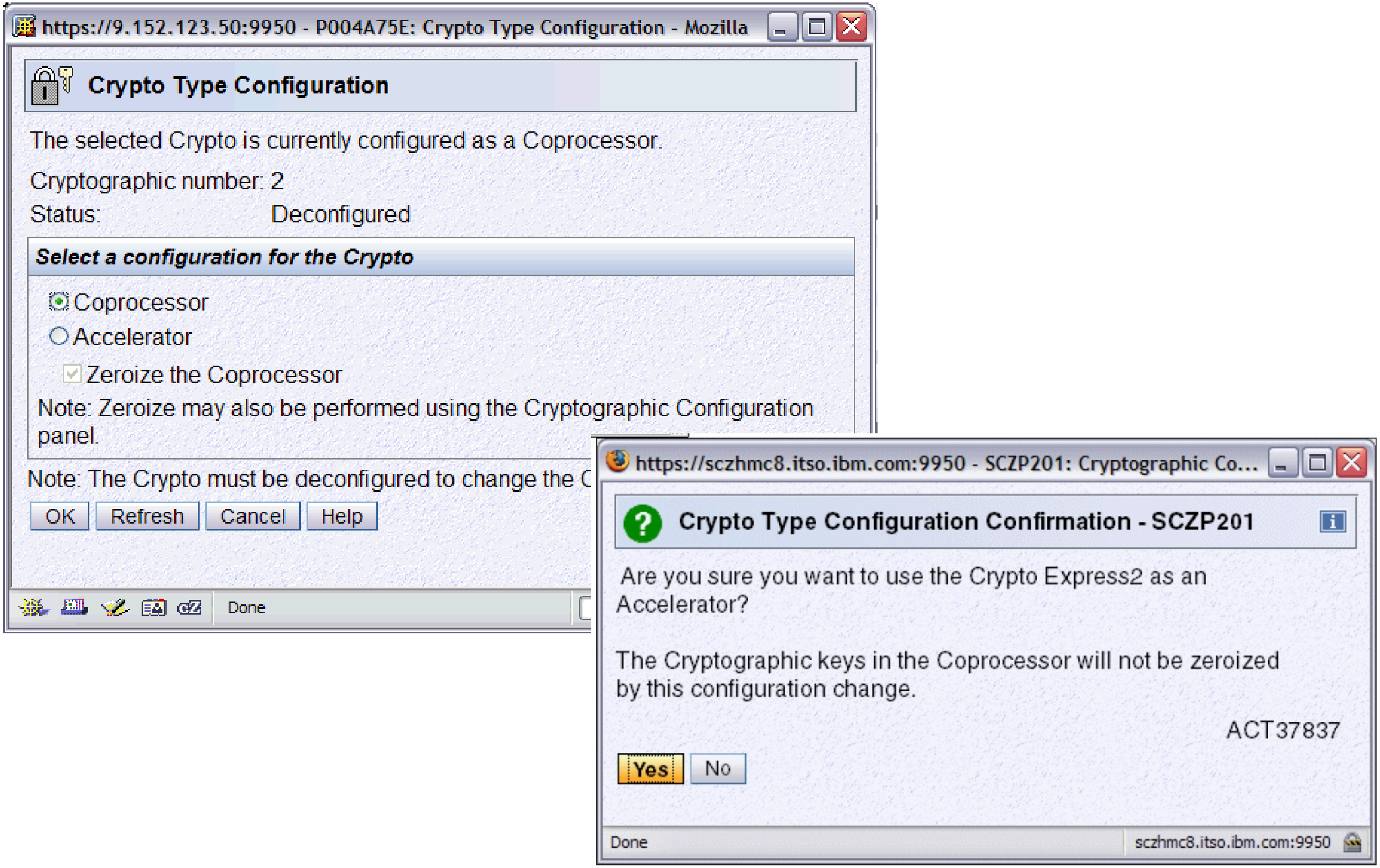
Cryptographic Configuration - SCZP201

Cryptographic Information

Select	Number	Status	Crypto Serial Number	Type	UDX Status	TKE Commands
○	0	Configured	97007984	X2 Coprocessor	IBM Default	Denied
○	1	Configured	97007985	X2 Coprocessor	IBM Default	Denied
○	2	Configured	97006647	X2 Coprocessor	IBM Default	Denied
○	3	Configured	97006644	X2 Coprocessor	IBM Default	Denied
○	4	Configured	97007866	X2 Coprocessor	IBM Default	Denied
○	5	Configured	97007853	X2 Coprocessor	IBM Default	Denied
●	6	Deconfigured	Not available	X2 Coprocessor	Not available	Not available
○	7	Configured	97006564	X2 Coprocessor	IBM Default	Denied

Select a Cryptographic number and then click the task push button.

LPAR hardware setup: CryptoExpress2 adapter



https://9.152.123.50:9950 - P004A75E: Crypto Type Configuration - Mozilla

Crypto Type Configuration

The selected Crypto is currently configured as a Coprocessor.

Cryptographic number: 2

Status: Deconfigured

Select a configuration for the Crypto

● Coprocessor

○ Accelerator

☑ Zeroize the Coprocessor

Note: Zeroize may also be performed using the Cryptographic Configuration panel.

Note: The Crypto must be deconfigured to change the C

OK
Refresh
Cancel
Help
Done

https://sczhmc8.itso.ibm.com:9950 - SCZP201: Cryptographic Co...

Crypto Type Configuration Confirmation - SCZP201

Are you sure you want to use the Crypto Express2 as an Accelerator?

The Cryptographic keys in the Coprocessor will not be zeroized by this configuration change.

ACT37837

Setup of Crypto Cards for z/VM guests

For shared crypto cards: Virtualization of crypto cards defined for the LPAR (all available cards which are not dedicated). These cards can be shared among guests

- Statement: CRYPTO APVIRT
- The guest gets one virtual card
- AP number and domain are chosen by VM and not identical with the one for the VM LPAR

For dedicated crypto cards: No virtualization of these crypto cards (not shared) will be done. Only the guest with such a definition can use these cards. The cards are not available to other guests

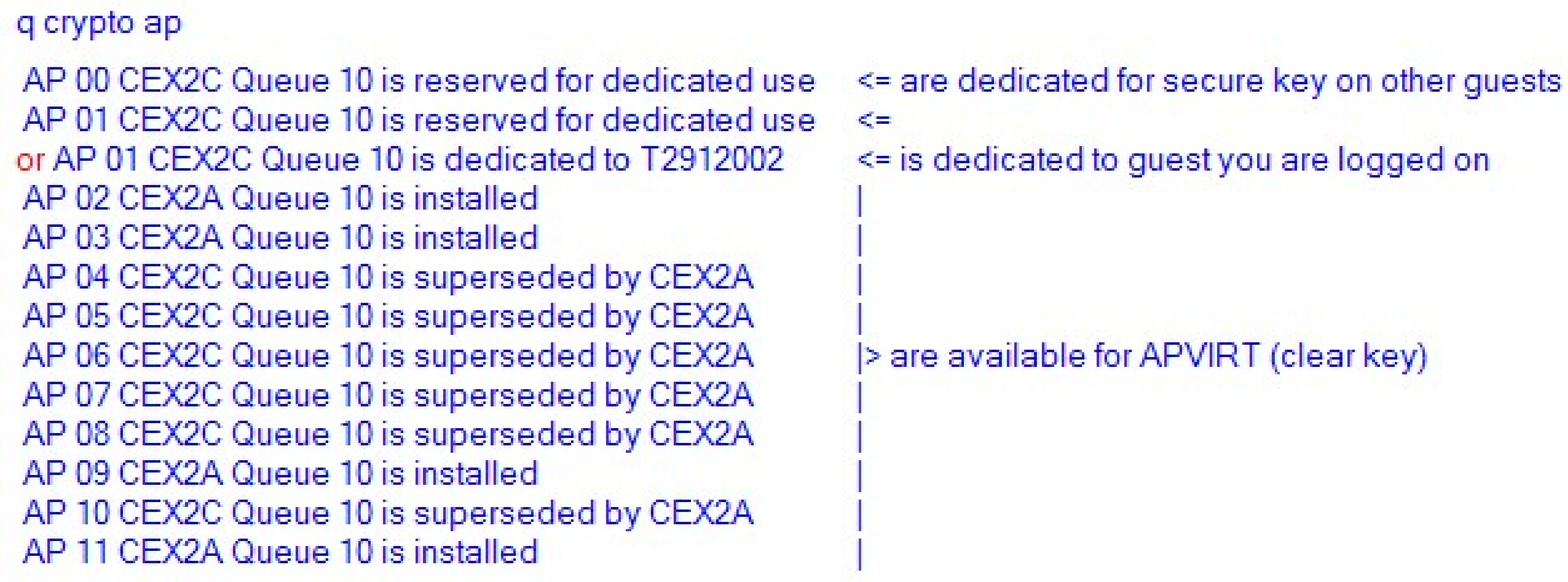
- Statement: CRYPTO DOMAIN x APDED y
- Domains: x can be one or more domains defined for the VM LPAR
- AP number: y can be one or more CEX2C cards defined for the VM LPAR

z/VM User Directory Example

```
USER LNXRH2 ****** 512M 1G G
INCLUDE IBMDFLT
CRYPTO APVIRT
IPL CMS PARM AUTOCR
MACHINE ESA 2
NICDEF C200 TYPE QDIO LAN SYSTEM
VSWITCH1
NICDEF C300 TYPE QDIO LAN SYSTEM
VSWITCH2
MDISK 0191 3390 0241 0080 LX9U1R MR
MDISK 0201 3390 0001 1000 LXC40B MR
MDISK 0202 3390 1001 9016 LXC40B MR
```

How to see cryptographic card definitions in z/VM

- `Q CRYPTO AP:` Shows the crypto cards and domains for the whole VM as defined in the LPAR activation profile

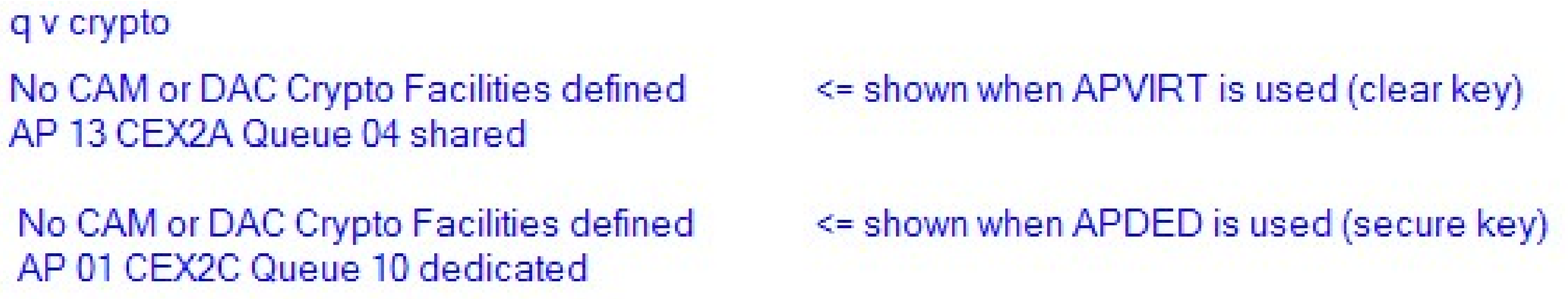


```
q crypto ap
 AP 00 CEX2C Queue 10 is reserved for dedicated use   <= are dedicated for secure key on other guests
 AP 01 CEX2C Queue 10 is reserved for dedicated use   <=
or AP 01 CEX2C Queue 10 is dedicated to T2912002      <= is dedicated to guest you are logged on
 AP 02 CEX2A Queue 10 is installed                    |
 AP 03 CEX2A Queue 10 is installed                    |
 AP 04 CEX2C Queue 10 is superseded by CEX2A          |
 AP 05 CEX2C Queue 10 is superseded by CEX2A          |
 AP 06 CEX2C Queue 10 is superseded by CEX2A          |> are available for APVIRT (clear key)
 AP 07 CEX2C Queue 10 is superseded by CEX2A          |
 AP 08 CEX2C Queue 10 is superseded by CEX2A          |
 AP 09 CEX2A Queue 10 is installed                    |
 AP 10 CEX2C Queue 10 is superseded by CEX2A          |
 AP 11 CEX2A Queue 10 is installed                    |
```

Note: AP 00 means crypto processor 00, equivalent to AP number in profile Queue 10 means crypto domain 10 as defined in profile

How to see cryptographic card definitions in z/VM

- Q V CRYPTO : Shows an entry with a queue statement and shows that the guest is connected to crypto cards

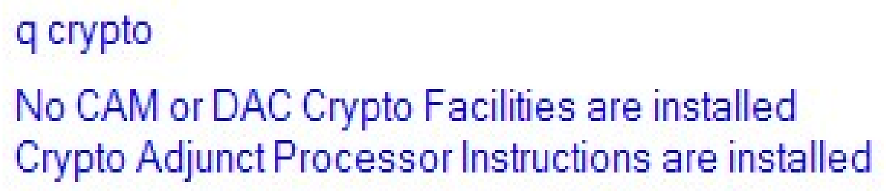


```
q v crypto

No CAM or DAC Crypto Facilities defined          <= shown when APVIRT is used (clear key)
AP 13 CEX2A Queue 04 shared

No CAM or DAC Crypto Facilities defined          <= shown when APDED is used (secure key)
AP 01 CEX2C Queue 10 dedicated
```

Note: shared : AP number and queue numbers are from virtulization

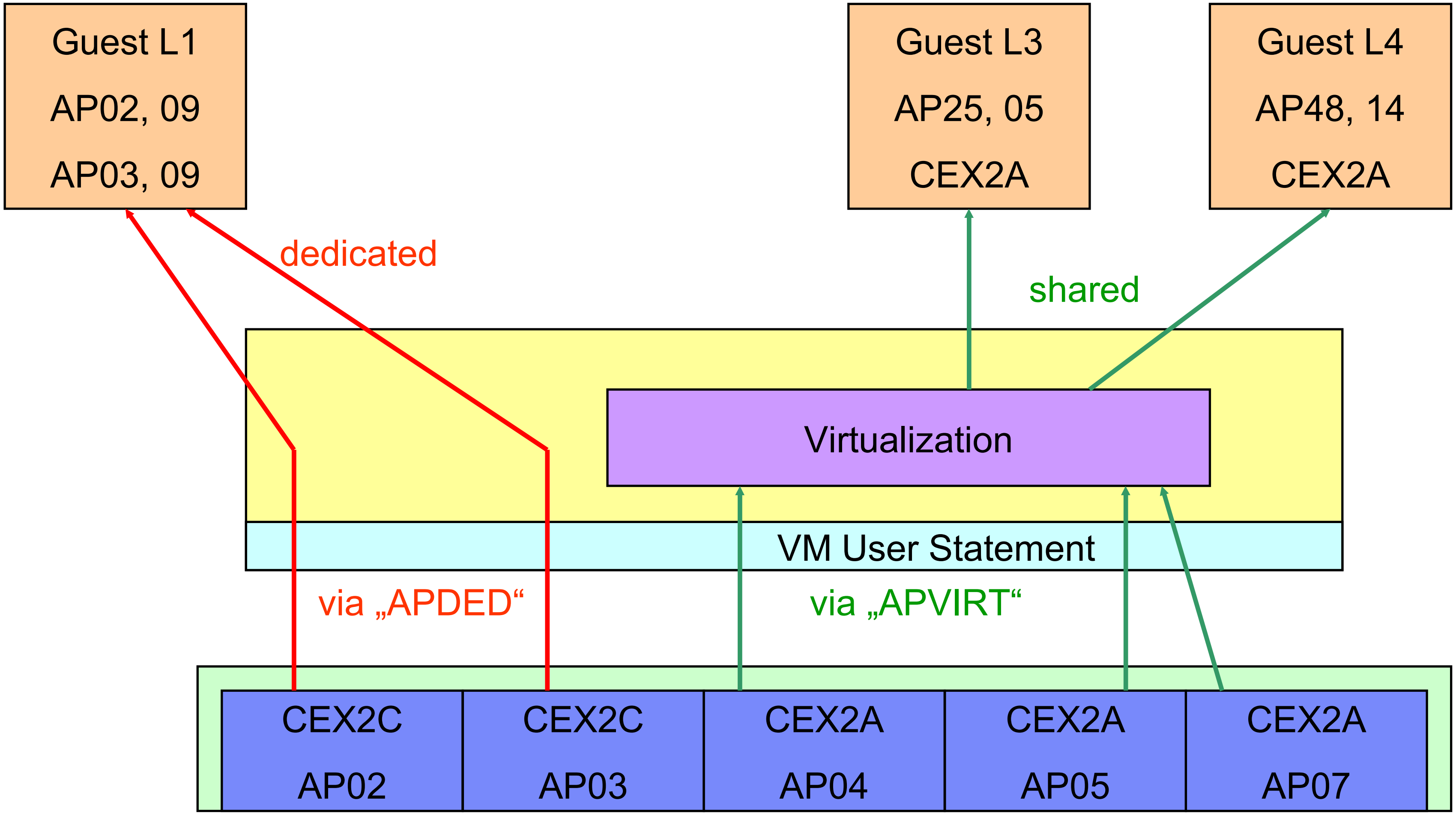
dedicated: AP nmuber and queue numbers are real values from the LPAR profile

- Shows that crypto cards (APs) can be used..

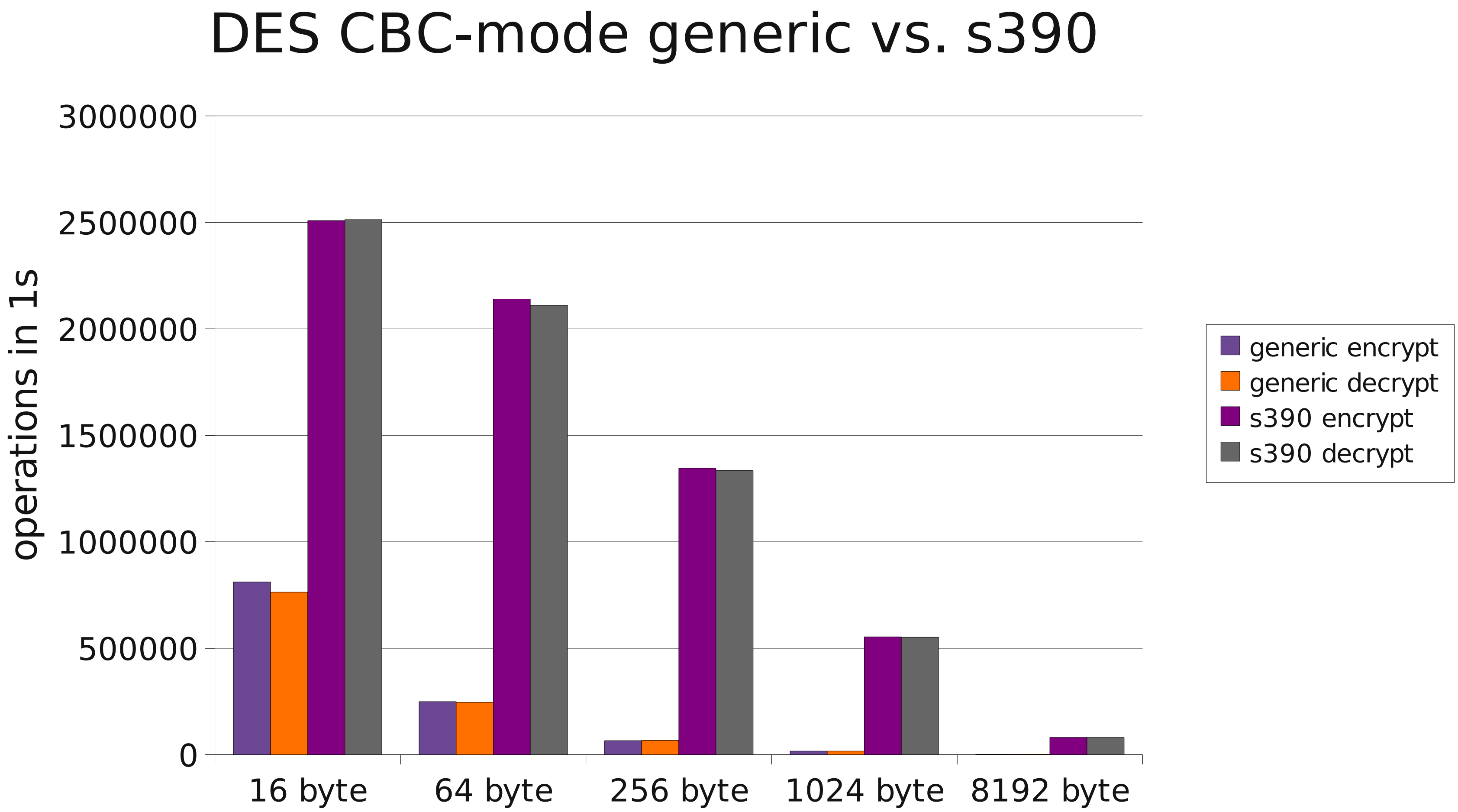


```
q crypto

No CAM or DAC Crypto Facilities are installed
Crypto Adjunct Processor Instructions are installed
```

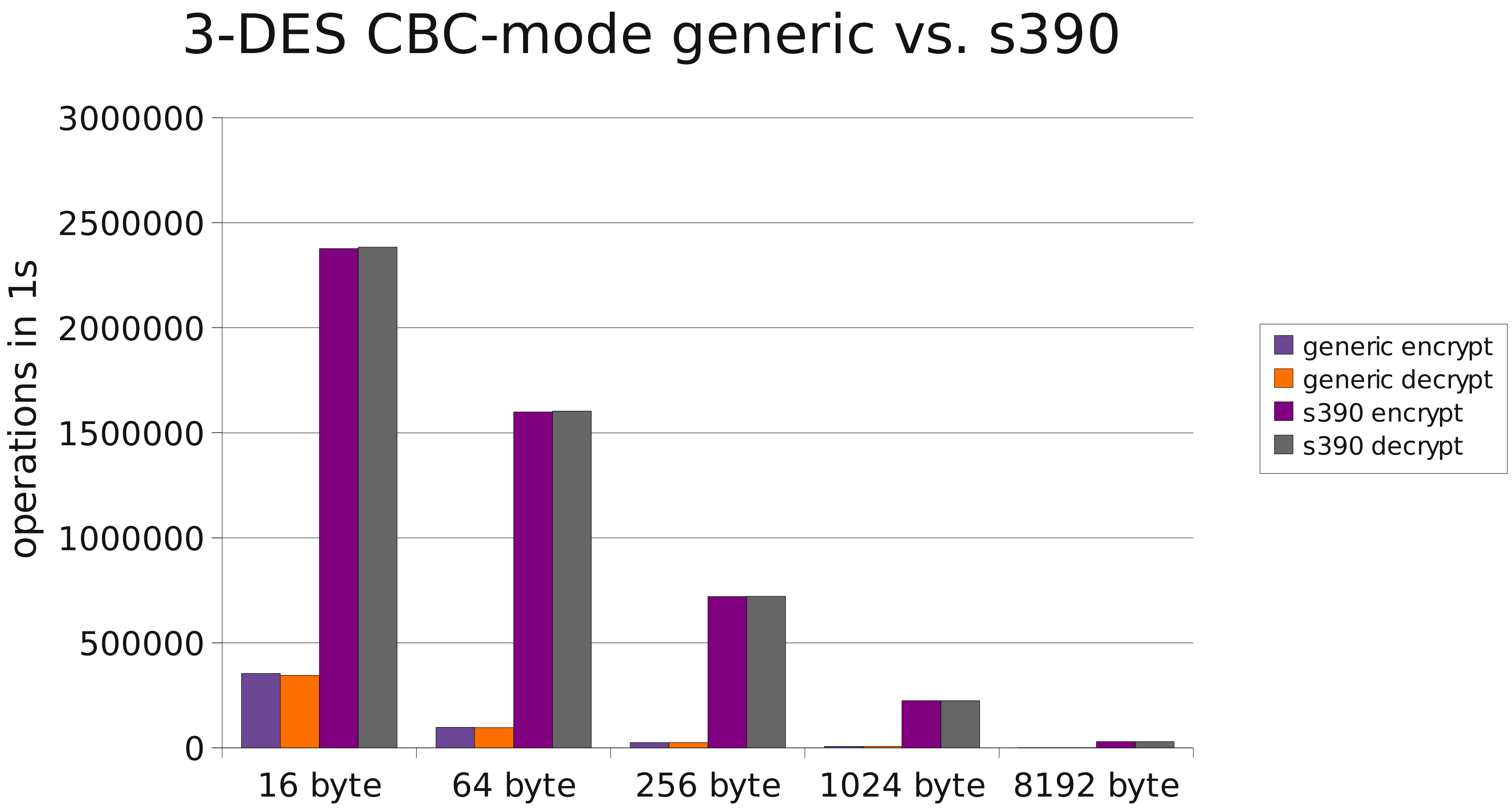
Dedicated and shared cards under VM



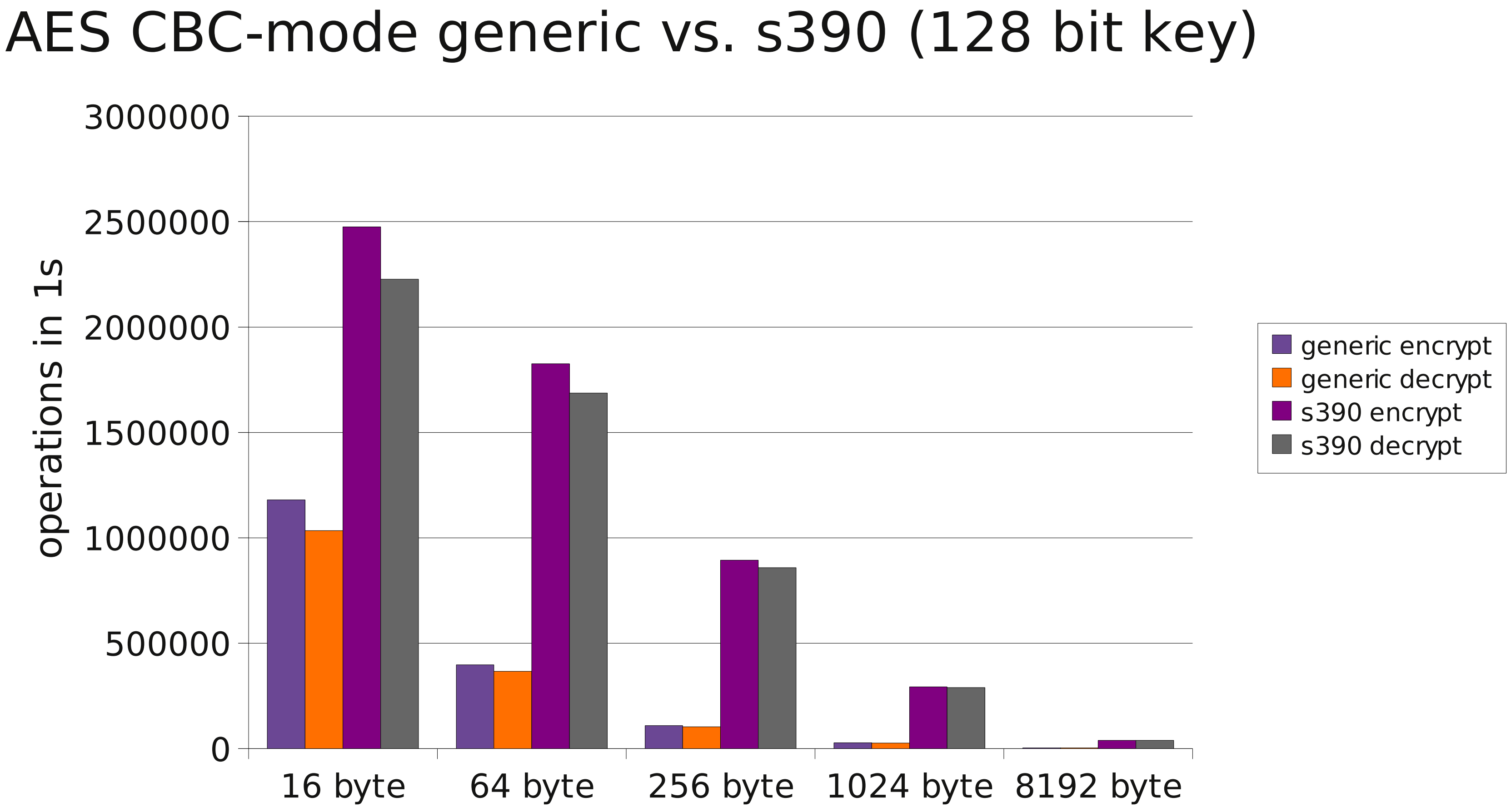
Performance: CPACF DES



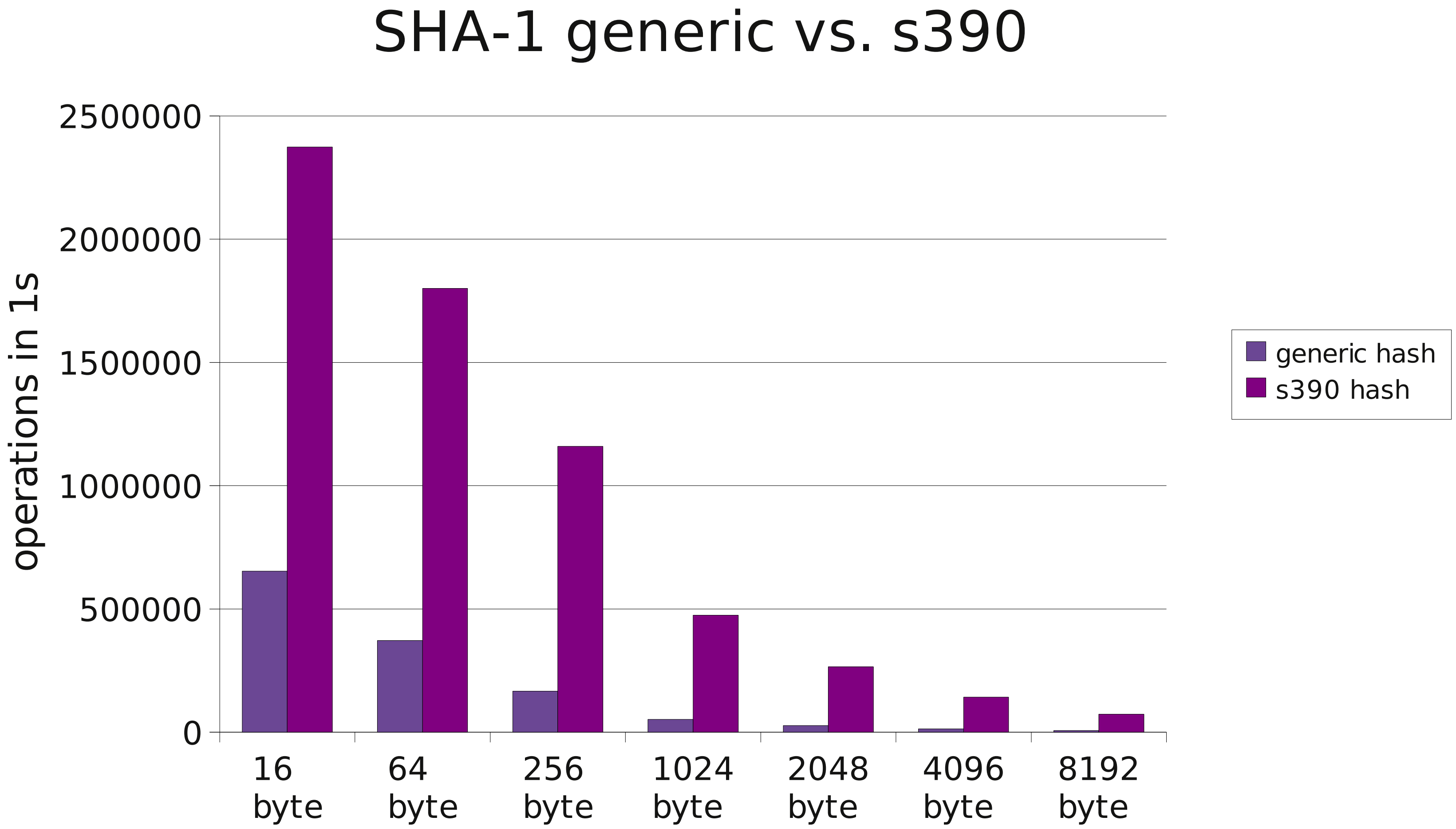
Performance: CPACF 3-DES



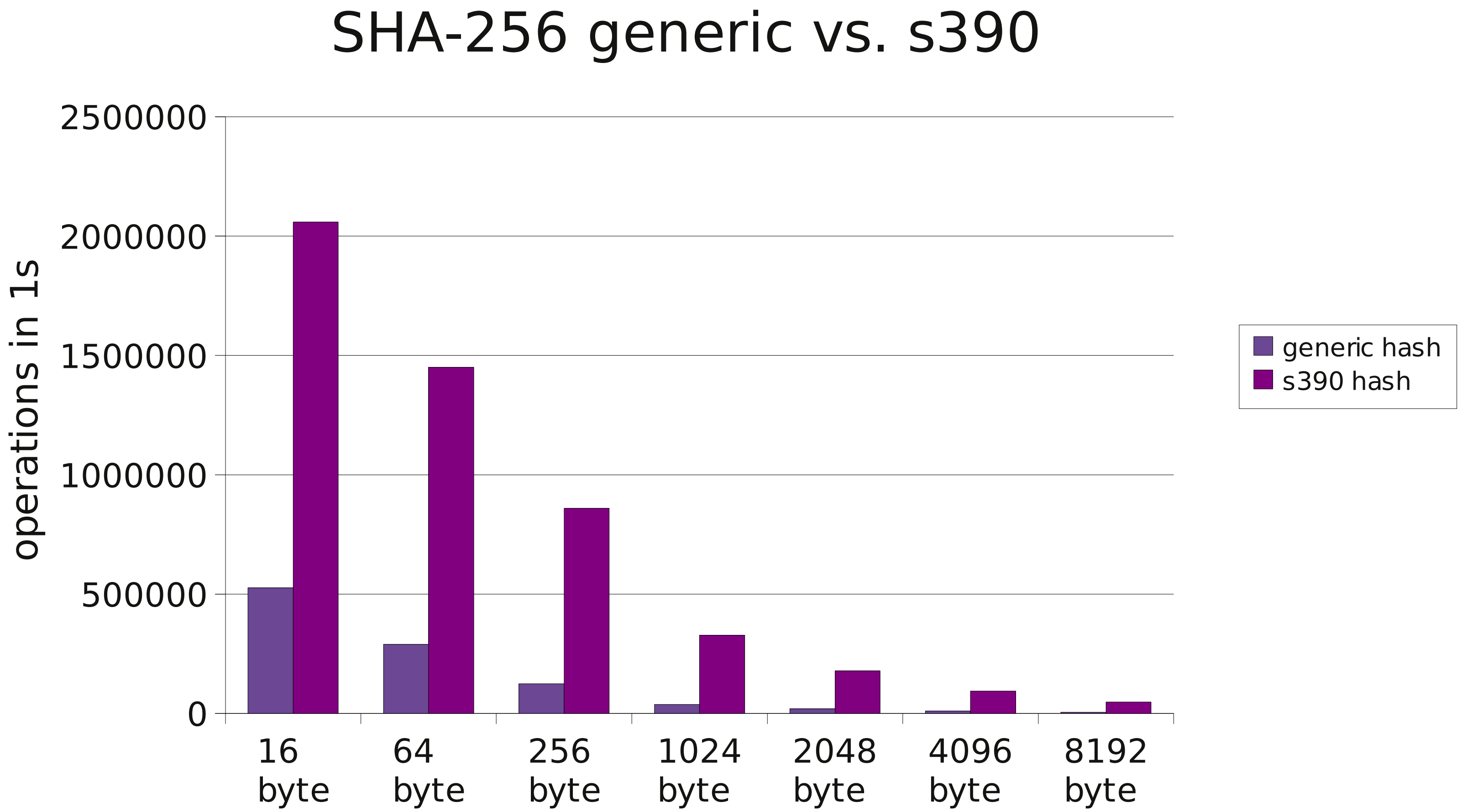
Performance: CPACF AES



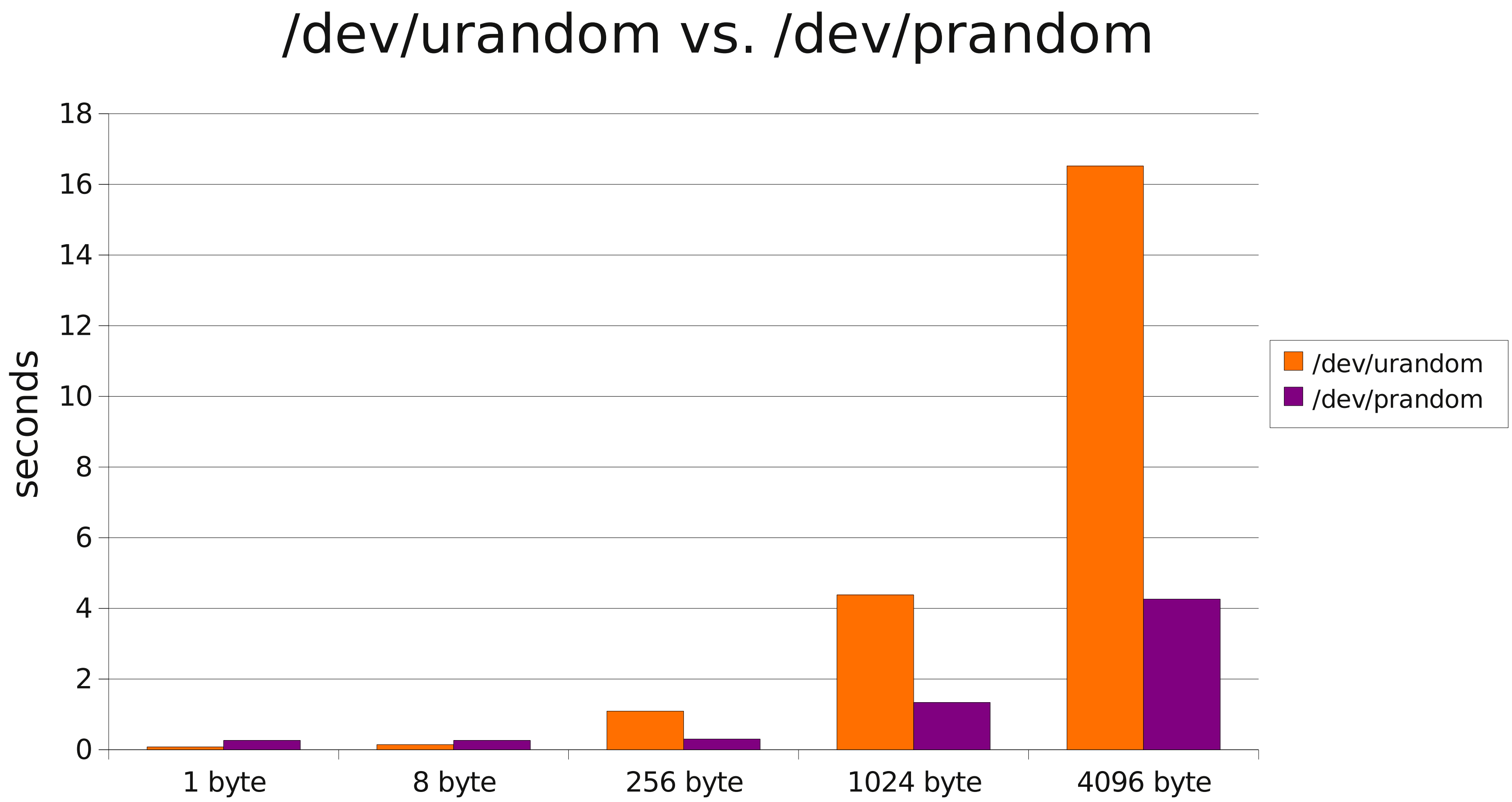
Performance: CPACF SHA-1



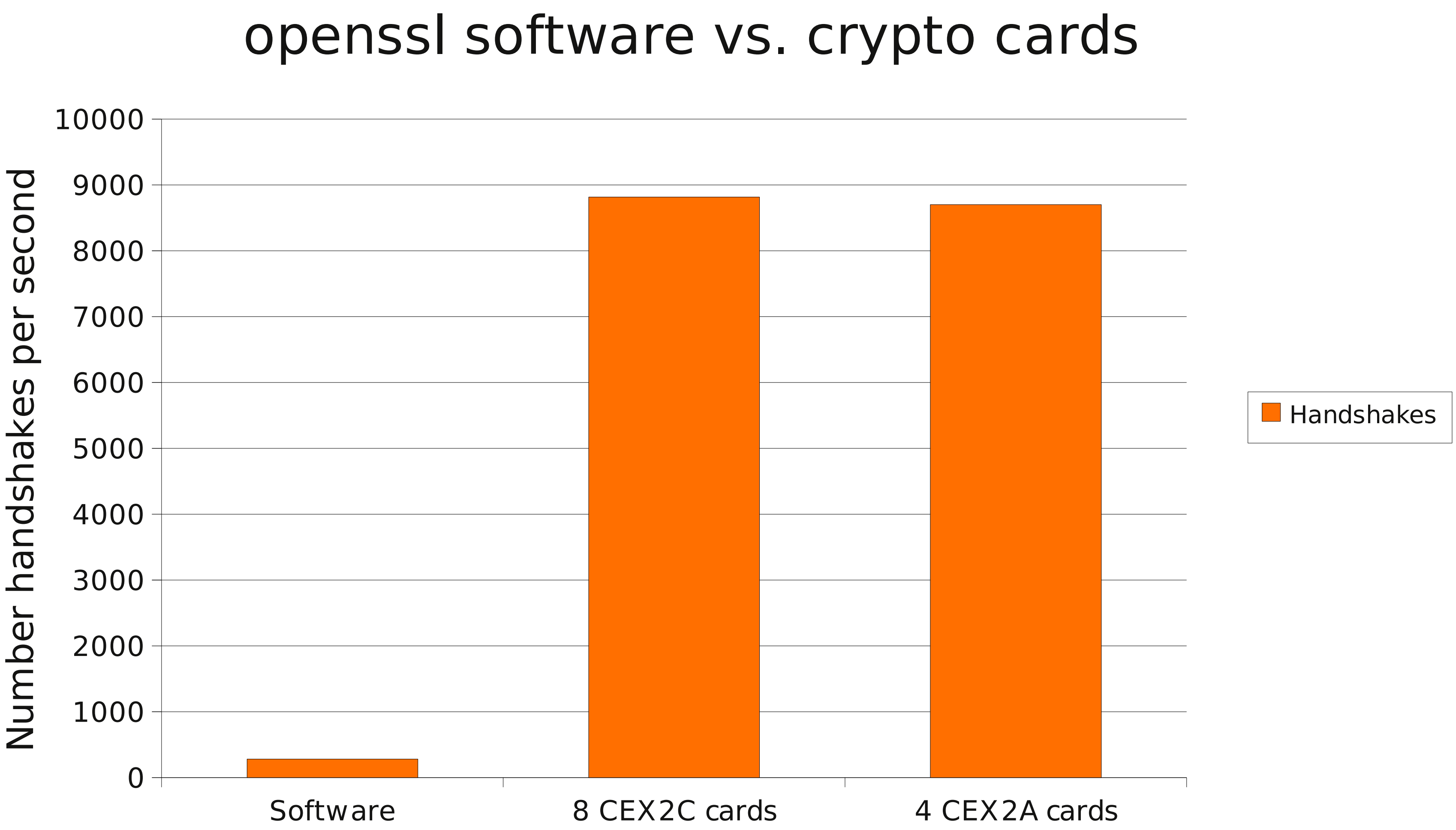
Performance: CPACF SHA-256



Performance: PRNG



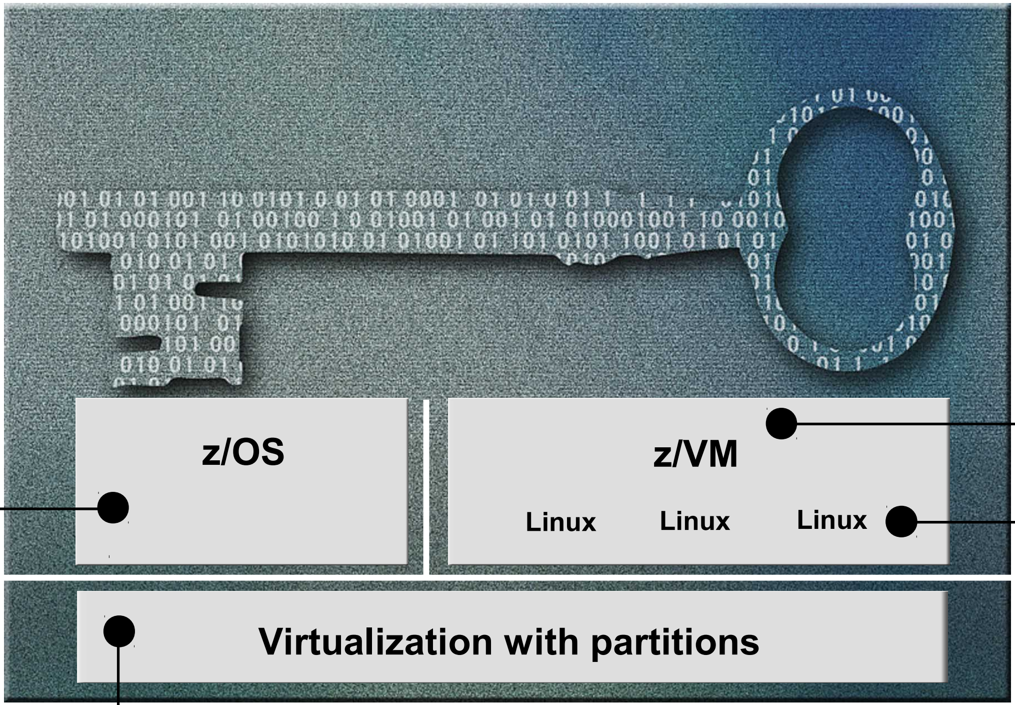
Performance: openssl



Certifications on System z

Security Server: RACF, LDAP, Firewall - Encryption - Public Key Infrastructures - Certificate Authority

The Common Criteria program developed by NIST and NSA establishes an organizational and technical framework to evaluate the trustworthiness of IT Products and protection profiles



z/VM

- **Common Criteria** EAL4+ with CAPP and LSPP
  - **z/VM 5.3 + RACF**

Linux on System z

- **Common Criteria** EAL4+ with CAPP and LSPP
  - **SUSE LES9 certified**
- **Common Criteria** EAL3+ with CAPP and LSPP
  - **Red Hat EL4 certified at EAL4+**
  - **Red Hat EL5 EAL4+ in progress**

z/OS

- **Common Criteria** EAL4+ with CAPP and LSPP
  - z/OS 1.9 + RACF
- **IdenTrust™** certification for z/OS PKI Services

System z EC and other System z servers

- **Common Criteria** EAL5 with specific Target of Evaluation
  - **Logical partitions**
- FIPS 140-2 level 4
  - Crypto Express 2 as coprocessor

See: www.**ibm.com**/security/standards/st_evaluations.shtml