

Secure your z/VSE network connections using OpenSSL

Ingo Franzki
ifranzki@de.ibm.com



Agenda

— Introduction

— Cryptography basics

- Encryption algorithms
- Encryption keys
- Diffie-Hellman versus RSA
- Elliptic Curve Cryptography

— Using cryptography with z/VSE

- Full tape encryption
- Encryption Facility for z/VSE
- SSL/TLS
- Hardware cryptography support on IBM Z
- OpenSSL
- What's new with z/VSE V6.2



Why secure VSE ?

— Prevent unauthorized access to VSE and data

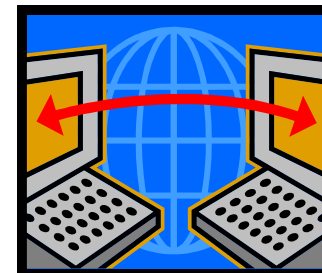
- Keep secret data secret
- Data modification by unauthorized users

— Prevent users from damaging the VSE system (maybe by accident)

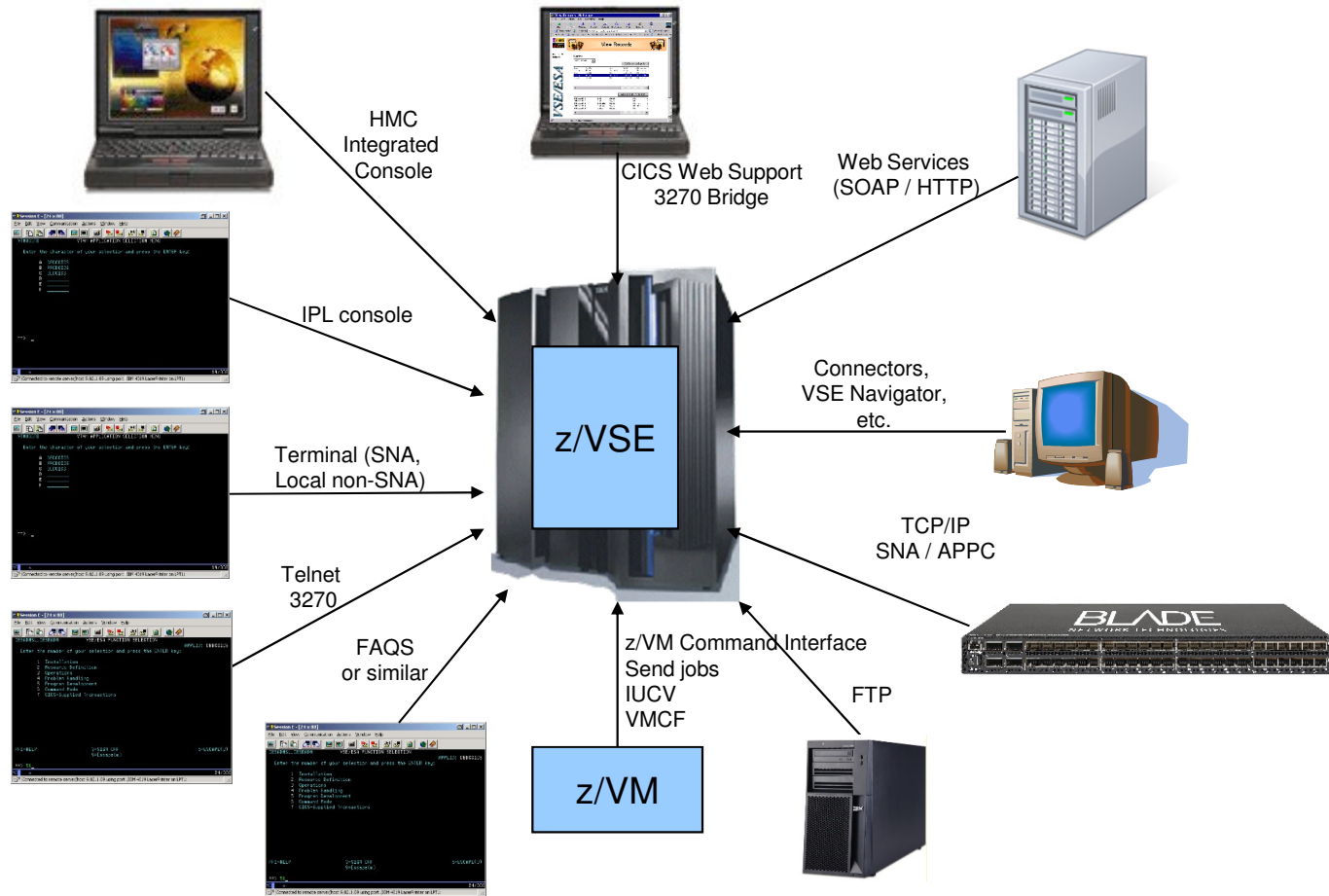
- Deletion of members or entries
- Submission of jobs

— Prevent unauthorized remote access to VSE

- Today most computers are part of a network
- Theoretically every system in the network could connect to your VSE system
- FTP allows to access production data
 - VSAM
 - POWER entries (listings)



Ways into your z/VSE system – Are you securing them all?



Agenda

— Introduction

— **Cryptography basics**

- Encryption algorithms
- Encryption keys
- Diffie-Hellman versus RSA
- Elliptic Curve Cryptography



— **Using cryptography with z/VSE**

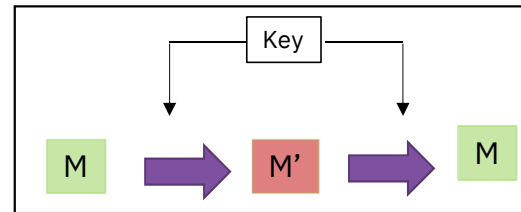
- Full tape encryption
- Encryption Facility for z/VSE
- SSL/TLS
- Hardware cryptography support on IBM Z
- OpenSSL
- What's new with z/VSE V6.2



Encryption basics

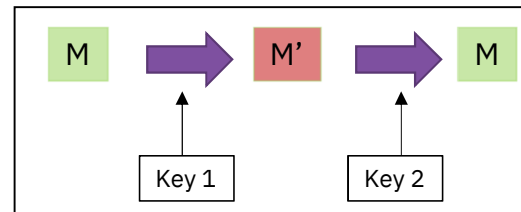
— Symmetric encryption

- The same key is used to encrypt and decrypt
- Example: RC4, DES, 3DES, AES



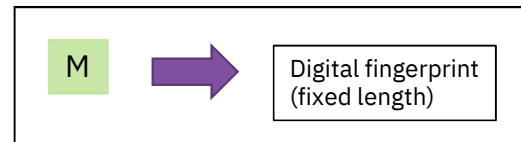
— Asymmetric encryption

- One key is used for encryption, another key is used for decryption (public and private keys)
- Example: RSA, Elliptic Curve Cryptography



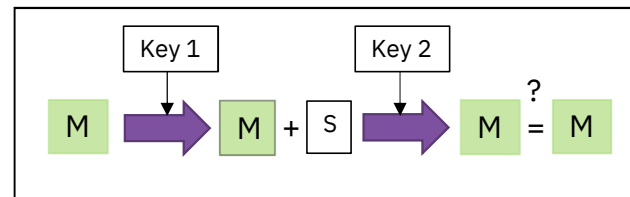
— Hash Algorithms

- A digital fingerprint of a text
- Example: MD5, SHA



— Signatures

- To create a digital signature asymmetric algorithms are used, mainly RSA



Different kinds of encryption keys

— Keys that consist of **numbers** which are based on mathematical algorithms (asymmetric algorithms)

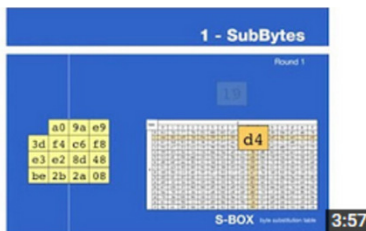
- RSA, Example: public key = (23,143), private key = (47,143)
 - Encryption of the number 7: $7^{23} \bmod(143) = 2$
 - Decryption: $2^{47} \bmod(143) = 7$

In this example, one could easily 'guess' the private key of 47 (i.e. brut force).

In reality this is done using much longer numbers, e.g. numbers of 4096 bits length

— Keys that consist of **random bit patterns** (symmetric algorithms)

- The key consist of a bit pattern of fixed length, e.g.
 - 16 Bytes = 128 bit results in $2^{128} = 3,4 \cdot 10^{38}$ possibilities
 - 32 Bytes = 256 bit results $2^{256} = 1,1 \cdot 10^{77}$ possibilities
- Example: Youtube: <https://www.youtube.com/watch?v=evjFwDRTmV0>



Animation of RIJNDEAL CIPHER : AES Encryption algorithm

HowTo

1 year ago • 2,497 views

This animation is made by Mr. Enrique Zabala. This is verison 4 made for CrypTool. This video is made for students so that they ...

Encryption key sizes

... and its security level

RSA	ECDH	Symmetric	Hash	Security (bits)
		RC4		<?
		DES	MD5	<?
			SHA-1	<80
1024	160			80
2048	224	TDES	SHA-224	112
3072	256	AES-128	SHA-256	128
4096				
7680	384	AES-192	SHA-384	192
15360	512	AES-256	SHA-512	256

Why all this different encryption algorithms?

— Asymmetric algorithms

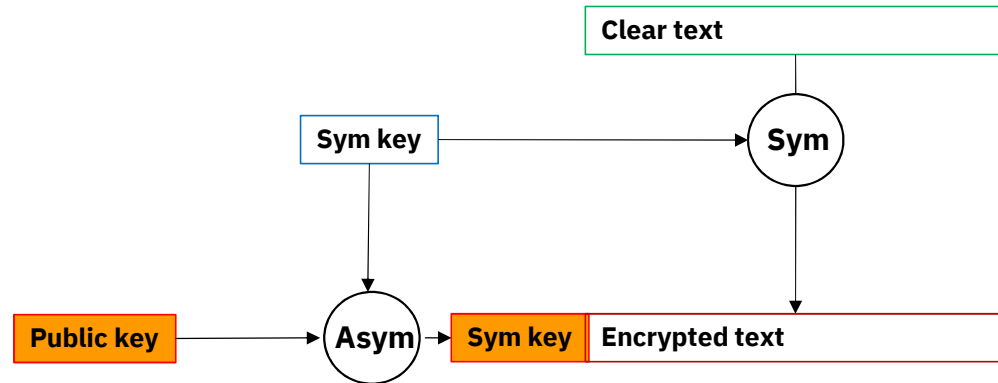
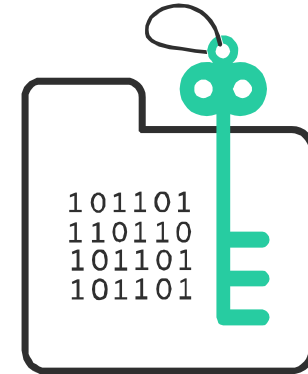
- Are slower by factors than symmetric algorithms
- Used to uniquely identify a communication partner
- Can only encrypt a certain number of bytes

— Symmetric algorithms

- Based on bit-shifting and logical computations (XOR, etc.)
- Very fast
- Can encrypt any numbers of bytes (usually in blocks of 8 or 16 bytes)

— Idea:

- Use symmetric algorithms to encrypt the data
- Use asymmetric algorithms to encrypt the symmetric key (session key)



Encryption modes (chaining)

— ECB (Electronic Codebook)

- Each data block is encrypted separately

— CBC (Cipher Block Chaining)

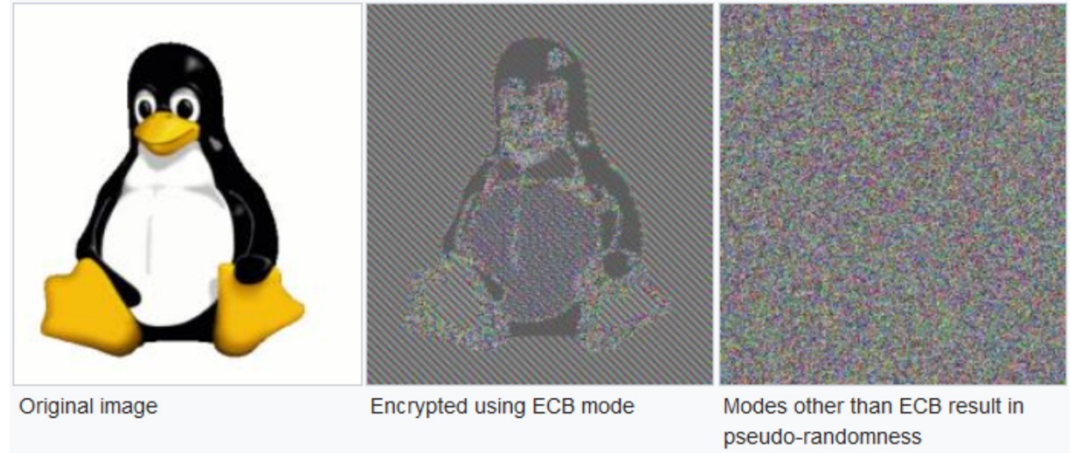
- The result of the encryption of one data block is fed into the encryption of the next data block

— GCM (Galois Counter Mode)

- Encryption and generation of a hash (digital fingerprint) in one step
- **Most current and securest mode**

— Others

- CFB - Cipher Feedback
- OFB - Output Feedback
- XTS - XEX-based tweaked-codebook mode with ciphertext stealing
- ...



Source: Wikipedia

SSL/TLS Connection establishment and key exchange

— RSA-based:

- Commonly used
- Long-term attacks are possible, because the session key is sent (encrypted) over the line

— Diffie-Hellman based:

- Usage increases
- Needs up to 30% more CPU
- Long-term, attacks are NOT possible (forward secrecy), because the session key is not sent over the line
- Usually used in combination with Elliptic Curve Cryptography (ECC)
- <https://www.youtube.com/watch?v=3QnD2c4Xovk>



Public Key Cryptography: Diffie-Hellman Key Exchange (short version)

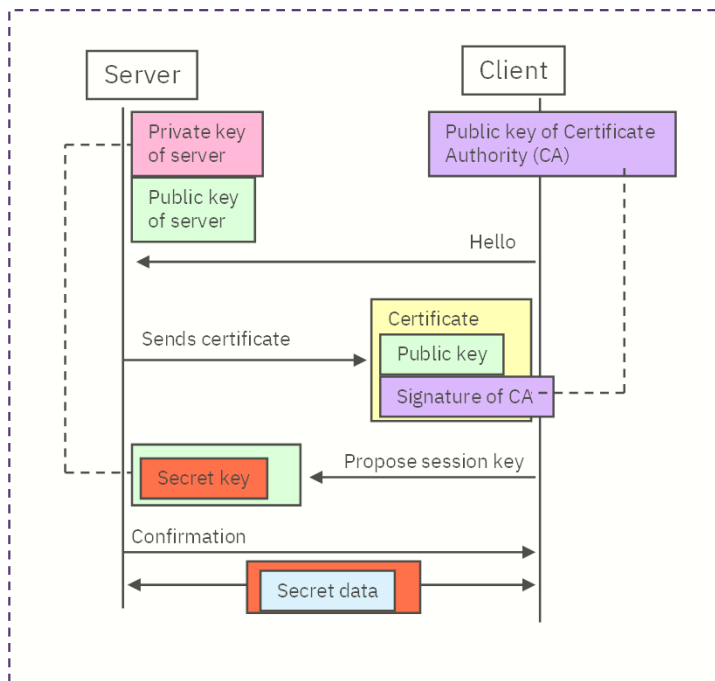
Art of the Problem

4 years ago • 366,437 views

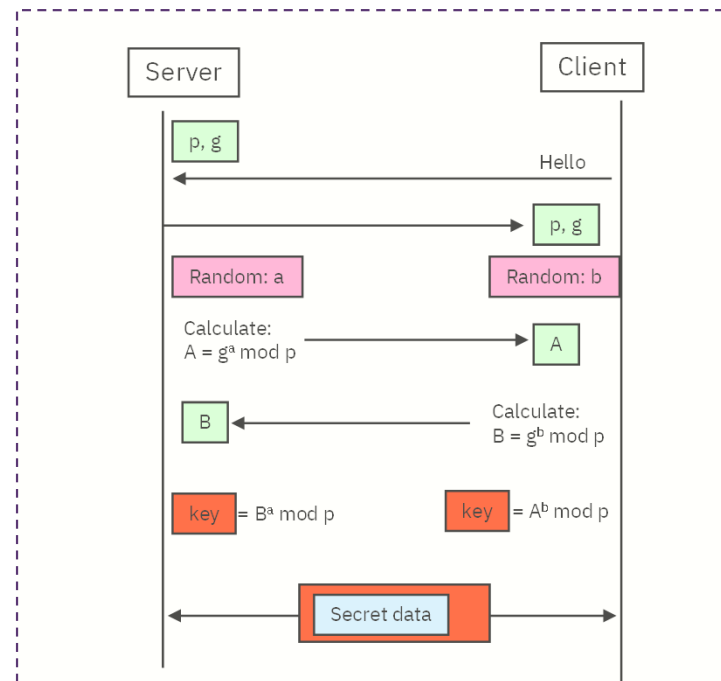
Diffie-Hellman key exchange was one of the earliest practical implementations of key exchange within the field of cryptography.

Diffie-Hellman versus RSA

RSA



Diffie-Hellman



➡ DH: session key does **not** go over the line

Diffie-Hellman versus RSA

— Diffie-Hellman

- Provides “forward secrecy”, because session key is not part of the session data
- Needs up to 30% more CPU
- Does not provide authentication, i.e. normally used together with RSA
- Often used together with Elliptic Curve Cryptography (ECC) for better performance
- Refer to Wikipedia / Youtube:
 - https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange
 - https://en.wikipedia.org/wiki/Elliptic_curve_Diffie%E2%80%93Hellman
 - <https://www.youtube.com/watch?v=YEBfamv-do>

— Special SSL/TLS cipher suites use Diffie-Hellman and Elliptic Curve

- DHE-RSA cipher suites: use Diffie-Hellman with RSA
- ECDHE-RSA cipher suites: use Diffie-Hellman with ECC and RSA

 DHE-RSA and ECDHE-RSA is supported on z/VSE with OpenSSL 1.0.1e or later

Some more info on Elliptic Curve Cryptography (ECC) ...

— Elliptic Curves

- Described through $y^2 = x^3 + ax + b$
- Mathematical calculation based on points on the curve

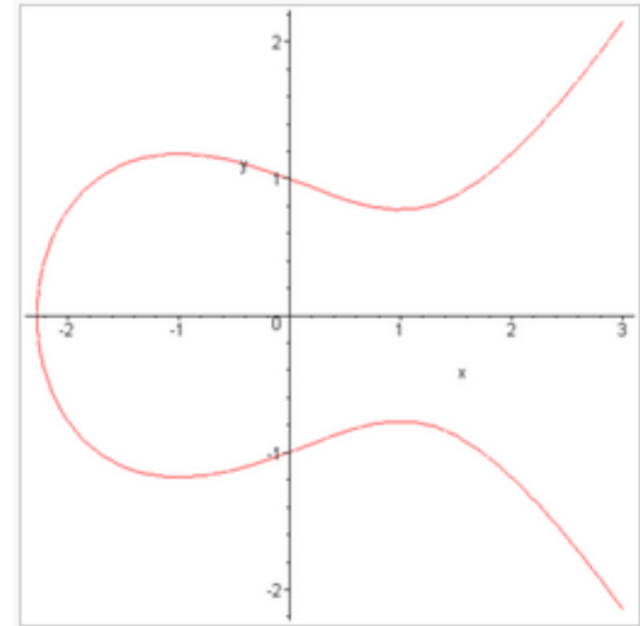
— Prime Curves (NIST)

— Brainpool curves

- Are being researched and provided by an working group of German governmental institutions and companies, including the German BSI (equivalent to U.S. NIST)
- Are supported with OpenSSL 1.0.2 (and Java)
- EC keys based on Brainpool curves are supported by Keyman/VSE
- Refer to
 - <http://www.ecc-brainpool.org/> (German website)
 - https://en.wikipedia.org/wiki/Elliptic_curve_cryptography#Implementation

— CEX4C, CEX5C, CEX6C

- Provide ECC acceleration
- **z/VSE 6.2: Added** Hardware acceleration for ECC in z/VSE and OpenSSL



Agenda

— Introduction

— Cryptography basics

- Encryption algorithms
- Encryption keys
- Diffie-Hellman versus RSA
- Elliptic Curve Cryptography

— Using cryptography with z/VSE



- Full tape encryption
- Encryption Facility for z/VSE
- SSL/TLS
- Hardware cryptography support on IBM Z
- OpenSSL
- What's new with z/VSE V6.2

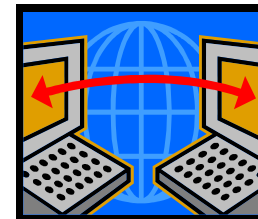


Using cryptography with z/VSE

Main areas of cryptography:

— Encryption of data transmitted over network connections (Data in flight)

- SSL/TLS, HTTPS
- SecureFTP

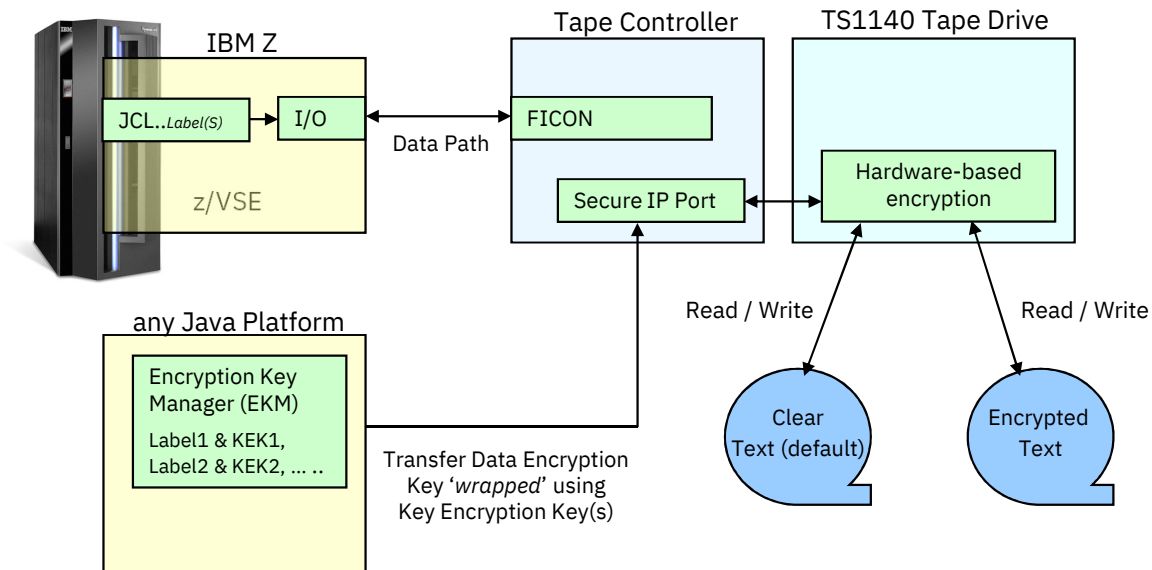


— Encryption of data stored on disk or tape (Data at rest)

- Encryption of backups or archives
- Exchange of encrypted and/or signed data with customers or business partners
- TS1140 Encrypting Tape Drive
- Encryption Facility for z/VSE



IBM Tape Encryption – TS1140



```

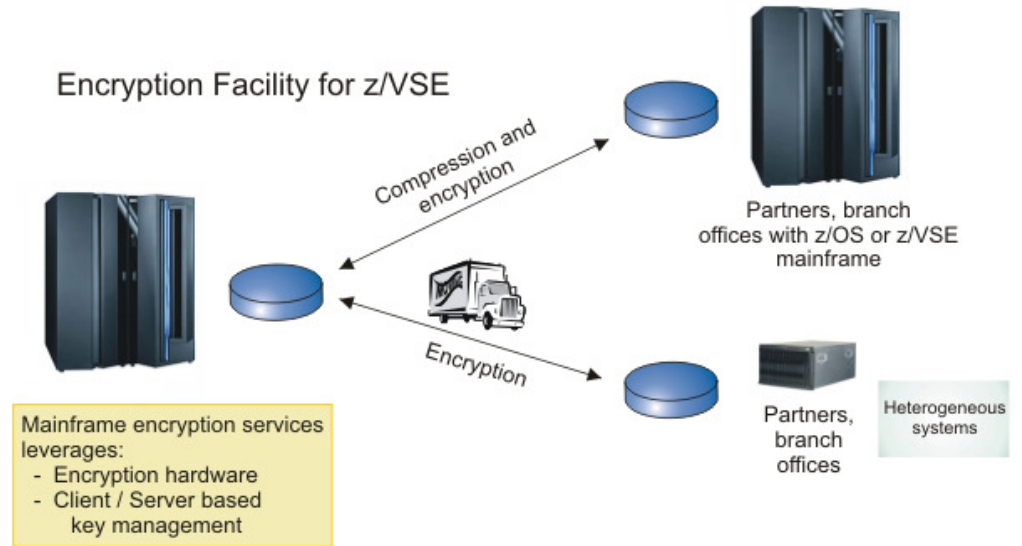
// JOB ENCRYPT
// ASSGN SYS005,480,03
// KEKL UNIT=480,KEKL1='MYKEKL1',KEM1=L,KEKL2='MYKEKL2',KEM2=L
// EXEC LIBR
  BACKUP LIB=PRD2 TAPE=SYS005
/*
/ &
  
```

Annotations:

- encryption mode (03=write):** Points to the value 03 in the ASSGN statement.
- key label1 (name of the 1. KEK-key in EKM):** Points to MYKEKL1 in the KEKL1 statement.
- encoding mechanism (L=Label, H=Hash):** Points to L in the KEM1 statement.

Encryption Facility for z/VSE

- Secure business and customer data
- Address regulatory requirements
- Protect data from loss and inadvertent or deliberate compromise
- Enable sharing of sensitive information across platforms with partners, vendors, and customers
- Enable **decrypting and encrypting of data** to be exchanged between z/VSE and non-z/VSE platforms



- The Encryption Facility for z/VSE is packaged as an **optional, priced feature** of z/VSE (5686-CF9-40).
- The **Encryption Facility for z/VSE V1.1** uses IBM Z data format
- The **Encryption Facility for z/VSE V1.2** uses the standard **OpenPGP** data format
 - PGP stands for „Pretty Good Privacy“, invented by Phil Zimmermann in 1991
 - Open Standard, described in RFCs 2440 and 4880
 - Compatible with Encryption Facility for z/OS V1.2 and many other OpenPGP implementations

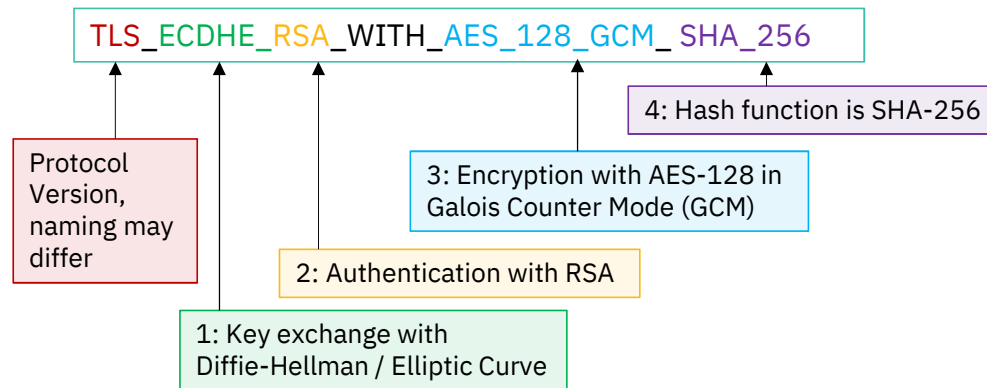
Transport Layer Security – Encrypted data transfer over a network

Formerly named SSL – Secure Socket Layer

- **TLS/SSL provides a communication channel with message integrity, authentication, and confidentiality**
- **TLS/SSL is a widely used protocol**
 - Secure HTTP (HTTPS) is used very often in the Internet
- **TLS/SSL uses a TCP connection to transfer encrypted messages**
 - Uses asymmetric cryptography for session initiating
 - Uses symmetric cryptography for data encryption
- **Cipher suites defines the algorithms used:**
 - For key exchange
 - For encryption
 - For hash algorithm

HTTP	App
TCP	
IP	

HTTP	App
TLS/SSL	
TCP	
IP	



Key & Certificate Management

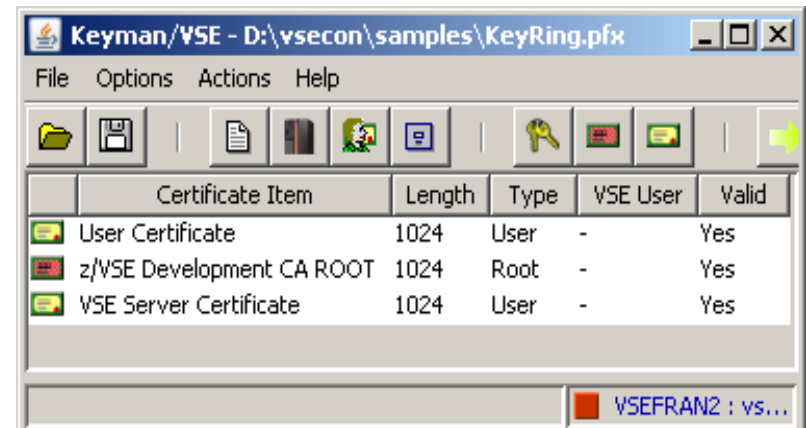
— Cryptography uses **Keys** and **Certificates**

— Key Management is not trivial

- Key must often be kept secure for a very long time
- You must be able to associate the encrypted data with the corresponding key(s)
- Encrypted data and the corresponding key(s) must be strictly separated

— Keyman/VSE

- Creation of RSA keys and digital certificates
- Upload of keys and certificates to VSE
- Creation of PKCS#12 keyring files (use with Java-based connector or import into a Web browser)
- Download from VSE Homepage
<https://www.ibm.com/it-infrastructure/z/zvse-downloads>



Keyman/VSE updates

— Elliptic Curve Crypto support

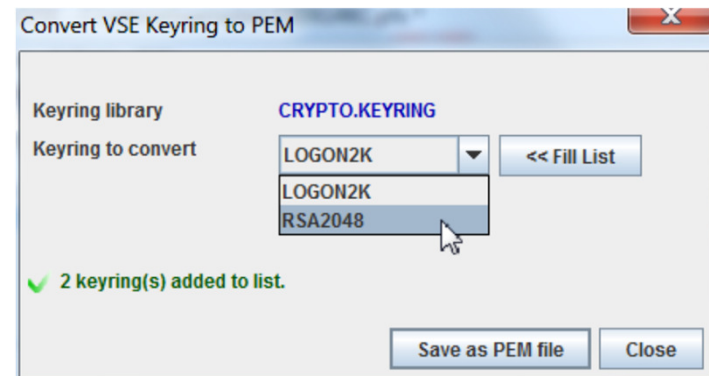
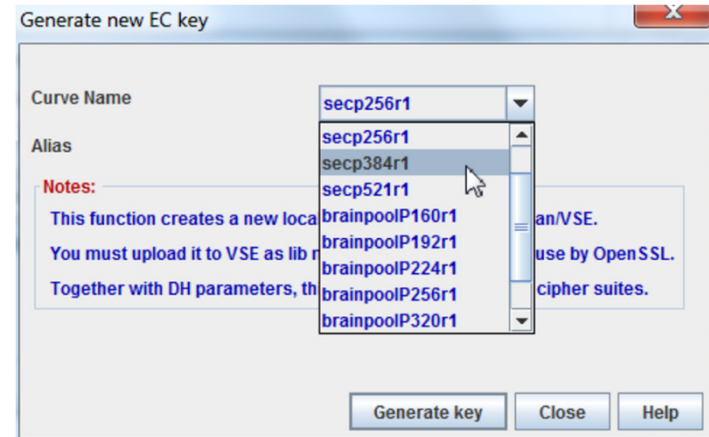
- Create and upload Elliptic-Curve (EC) key pairs.

— SHA-256 support

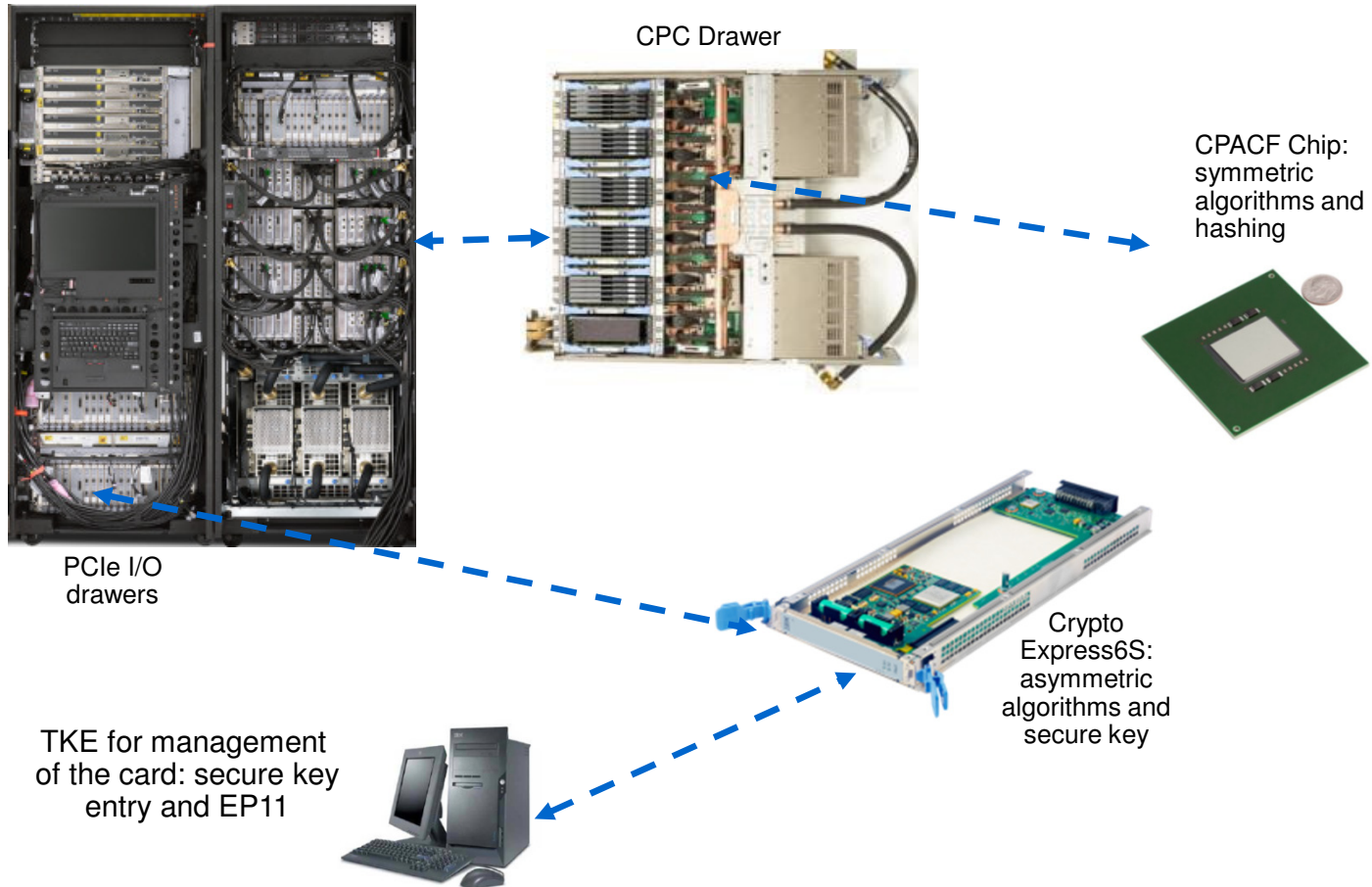
- Support SHA-256 signatures in certificates.
- This may require additional 1.5F zaps on TCP/IP for VSE.

— Convert CSI keyrings to OpenSSL PEM

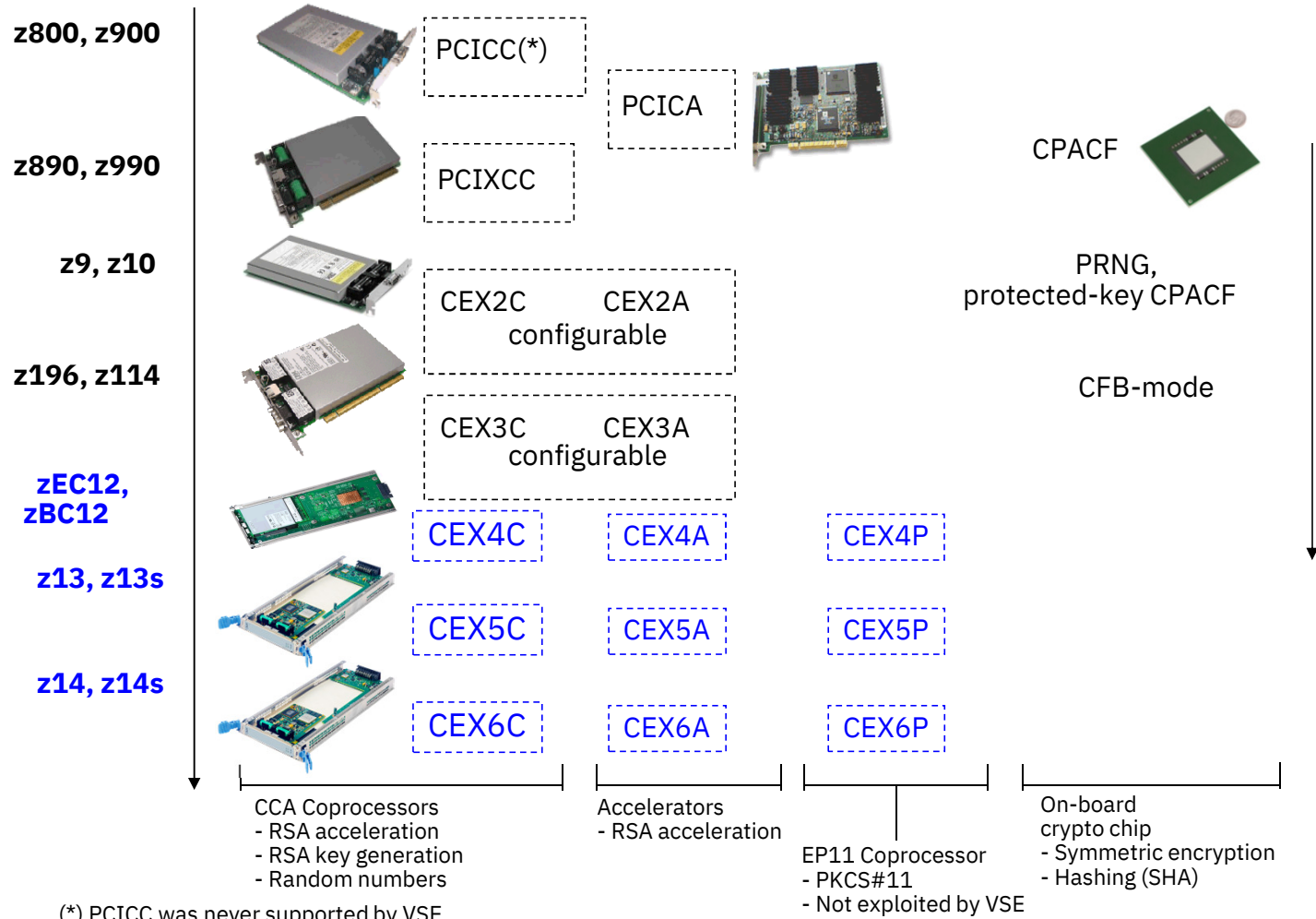
- Use existing PRVK, CERT, and ROOT members on VSE and build an equivalent PEM file for OpenSSL



Hardware Crypto Support on IBM Z



Hardware Crypto Support on IBM Z



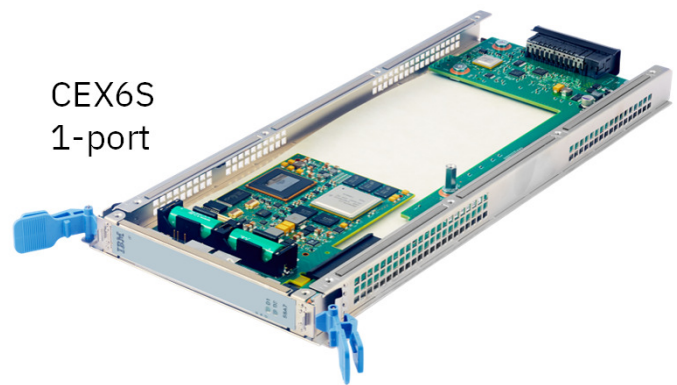
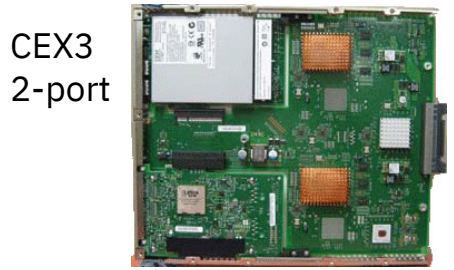
(*) PCICC was never supported by VSE

Crypto Express6S

- **Exclusive to IBM z14**
- **One-port card, i.e. one AP (adjunct processor) per physical card**
 - 2 cards min, 16 cards max per machine
- **Seneca I/O cage (the ‘S’ in the name)**
- **Can be configured in one of three ways:**
 - CEX6A: Accelerator
 - CEX6C: IBM Common Cryptographic Architecture (CCA) coprocessor
 - CEX6P: IBM Enterprise Public Key Cryptography Standards (PKCS) #11 (EP11) coprocessor
- **Form factor comparison CEX3 / CEX6S:**

Support for Crypto Express6S

- Included in **z/VSE 6.2** GA version
- APAR DY47715 for z/VSE 5.2
- APAR DY47716 for z/VSE 6.1



z/VSE Hardware Configuration

— z/VSE hardware configuration not necessary for crypto hardware

- No IOCDS definition in VSE
- No device type
- No ADD statement
- You may have to define the devices in the HMC (LPAR) or z/VM directory



— Use of crypto hardware is transparent to end users and applications

- But use of crypto hardware can be disabled via option

— How to setup cryptographic hardware for VSE:

- https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/technical_articles_and_whitepapers.html

```
FB 0095 1J054I FOUND A CRYPTO EXPRESS5S CARD AT AP 0  
FB 0095 1J054I FOUND A CRYPTO EXPRESS5S CARD AT AP 3  
FB 0095 1J005I HARDWARE CRYPTO DEVICE DRIVER INITIALIZED SUCCESSFULLY.  
FB 0095 1J006I USING AP QUEUE 79
```

OpenSSL Support

— What is OpenSSL?

- OpenSSL is an Open Source project providing an SSL/TLS implementation and key management utilities
- Available for most Unix-style operating systems, MAC, Windows, and IBM System i (OS/400)
- For details on OpenSSL refer to <http://www.openssl.org/>

— What is available on z/VSE?

- OpenSSL 1.0.2h runtime library
- Separate component: z/VSE cryptographic services, 5686-VS6-17-562
- Software implementations for all algorithms with all key lengths
- Hardware Crypto Support (Crypto Express cards and CPACF)
- Programming APIs:
 - OS390 / z/OS compatible SSL API (gsk_initialize(), gsk_secure_soc_init(), etc.)
 - Subset of the OpenSSL API (LE/C)



SSL/TLS related news with z/VSE 6.2

— **OpenSSL component of z/VSE enhancements:**

- The OpenSSL component of z/VSE (z/VSE Cryptographic Services) was upgraded to benefit from newer SSL/TLS functions
- The OpenSSL component transparently uses hardware acceleration for Elliptic Curve Cryptography (ECC), if available

— **CICS TS V2.2 security enhancements:**

- OpenSSL support for CICS Web Support will give clients more flexibility and allow them to take advantage of the OpenSSL security.
- Enable OpenSSL via // SETPARM BPX\$GSK="IJBGSKOS"

— **EZA API enhancements:**

- The EZA 'Multiplexer' and the EZA OpenSSL support will simplify the use of the EZA interface with any TCP/IP stack and allow to transparently use OpenSSL with EZA SSL-applications

— **VTAPE enhancements:**

- Clients can use SSL/TLS connections for remote VTAPES (virtual tapes) to protect sensitive data during network transfer

z/VSE 6.2: EZA Multiplexer

— With the EZASOKET and EZASMI interfaces you can specify which socket interface module to the TCP/IP partition is to be used

- Default: EZASOH99 (for TCP/IP for z/VSE)

— Select the EZA socket interface routine:

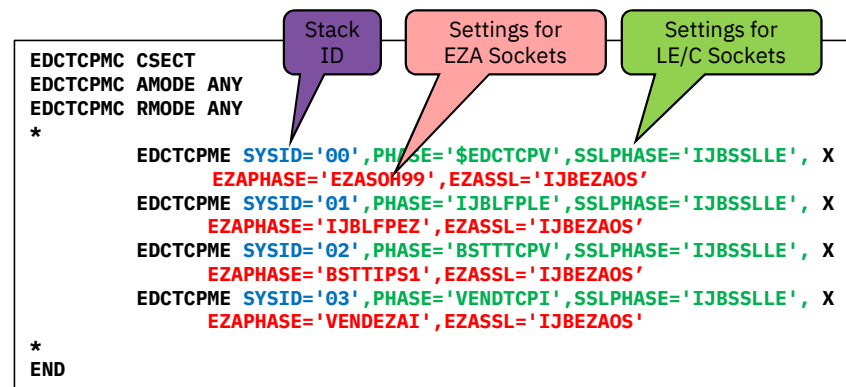
- Via JCL statement: // SETPARM [SYSTEM,]EZA\$PHA='phasename'
- Via parameter ADSNAME on the EZAAPI/EZASOKET INITAPI call

— The EZA Multiplexer can be used to ease the correct setup of socket interface modules for the corresponding stack IDs

- The multiplexer allows you to perform a one time setup and to assign the corresponding socket interface modules to the stack IDs
- The use of the multiplexer is transparent for your application

— Setup:

- Select IJBEZAMX as EZA interface phase for all applications (via SETPARM SYSTEM)
- Configure Multiplexer via EDCTCPMC in ICCF library 62
 - Same configuration as for LE/C Socket API Multiplexer
 - Additional parameters now allow to specify EZA interfaces



z/VSE 6.2: EZA OpenSSL Support

— Besides the EZA socket interface routine, The EZA Multiplexer also allows you to specify

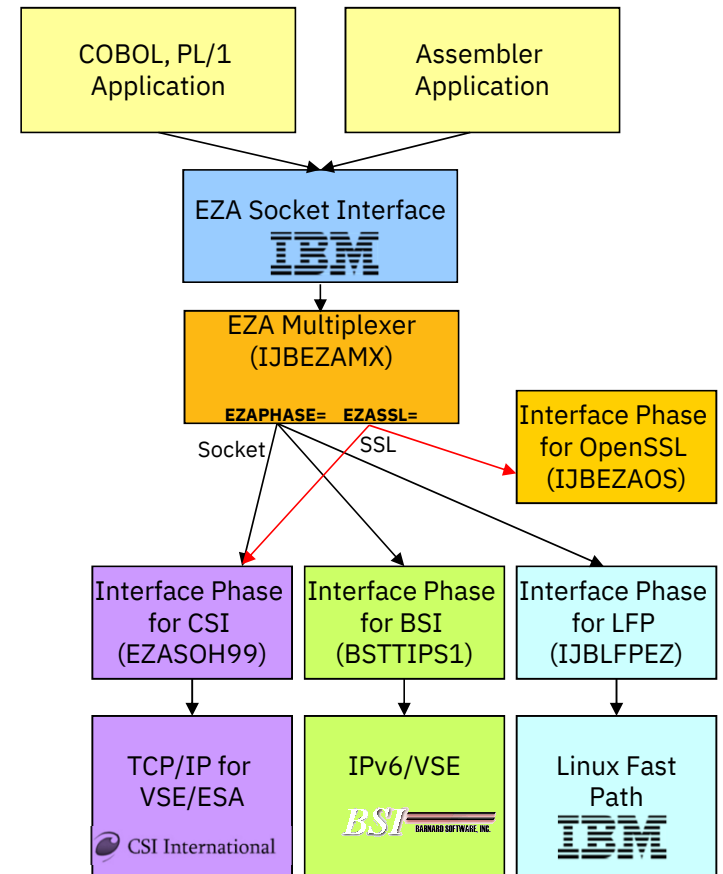
- The EZA socket interface routine
- an alternative EZA SSL interface routine
 - Default: The same as the EZA socket interface routine

— The new EZA SSL interface routine IJBEZAOS provides an interface to z/VSE's OpenSSL implementation

— The use of an alternative EZA SSL interface routine is transparent for your application

- OpenSSL uses different key and certificate formats (e.g .PEM instead of .PRVK, .ROOT, .CERT)

→ This makes z/VSE's OpenSSL support available for non-LE/C applications (i.e. COBOL, PL/1, HLASM)



OpenSSL Exploitation

— CICS TS for z/VSE V2.2: OpenSSL for CICS Web Support

- Enable OpenSSL via // SETPARM BPX\$GSK="IJBGSKOS"

— z/VSE Connectors

- z/VSE Connector Server (LE/C)
- z/VSE Script Server & Client (LE/C)
- z/VSE Database Connector/DBCLI (EZA)
- z/VSE HTTP Client (LE/C)
- z/VSE SOAP Engine
 - As server: OpenSSL for CICS Web Support
 - As client: z/VSE HTTP Client (LE/C)
- z/VSE REST Engine
 - As server: OpenSSL for CICS Web Support
 - As client: z/VSE HTTP Client (LE/C)
- z/VSE LDAP Client and Signon support (LE/C)
- z/VSE VTape Tape Data Handler (LE/C)

— IPv6/VSE product

- SSL Proxy Server (BSTTPRXY)
- Automatic TLS Facility (BSTTATLS)

— Applications using the LE/C socket interface:

- Enable via LE/C socket API multiplexer

— Applications using the EZA socket interface:

- Enable via EZA socket API multiplexer

Enable OpenSSL for Applications

LE/C Applications

— Configure the LE/C Socket API Multiplexer

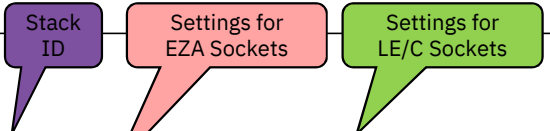
- via EDCTCPMC in ICCF library 62
- Set **SSLPHASE='IJBSLLE'** for the desired SYSID

EZA Applications

— Configure the EZA Socket API Multiplexer

- via EDCTCPMC in ICCF library 62
- Set **EZASSL='IJBEZAOS'** for the desired SYSID

```
EDCTCPMC CSECT
EDCTCPMC AMODE ANY
EDCTCPMC RMODE ANY
*
EDCTCPME SYSID='00',PHASE='$EDCTCPV',SSLPHASE='IJBSLLE', X
EZAPHASE='EZASOH99',EZASSL='IJBEZAOS'
EDCTCPME SYSID='01',PHASE='IJBLFPLE',SSLPHASE='IJBSLLE', X
EZAPHASE='IJBLFPEZ',EZASSL='IJBEZAOS'
EDCTCPME SYSID='02',PHASE='BSTTTCPV',SSLPHASE='IJBSLLE', X
EZAPHASE='BSTTIPS1',EZASSL='IJBEZAOS'
EDCTCPME SYSID='03',PHASE='VENDTCPI',SSLPHASE='IJBSLLE', X
EZAPHASE='VENDEZAI',EZASSL='IJBEZAOS'
*
END
```

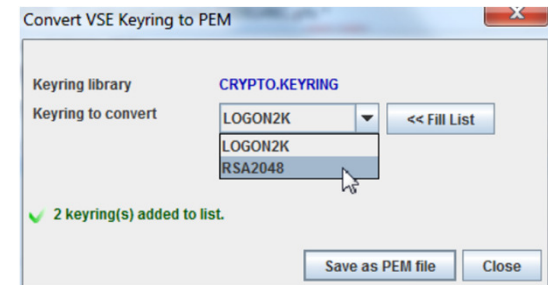


CICS Web Support

— Enable OpenSSL via // **SETPARM [SYSTEM] BPX\$GSK='IJBGSKOS'**

Keys and certificates must be converted into OpenSSL's PEM format

— Keyman/VSE supports converting a CSI keyring into a PEM keyring



Questions?



THANK YOU

Notices and disclaimers

- © 2019 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- **U.S. Government Users Restricted Rights – use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those
- customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers continued

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of 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. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**
- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.
- IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml