MAINFRAME CRYPTO

Unscrambling the Complexity of Crypto!

Transporting Crypto Keys

gregboyd@mainframecrypto.com

THE Z EXCHANGE

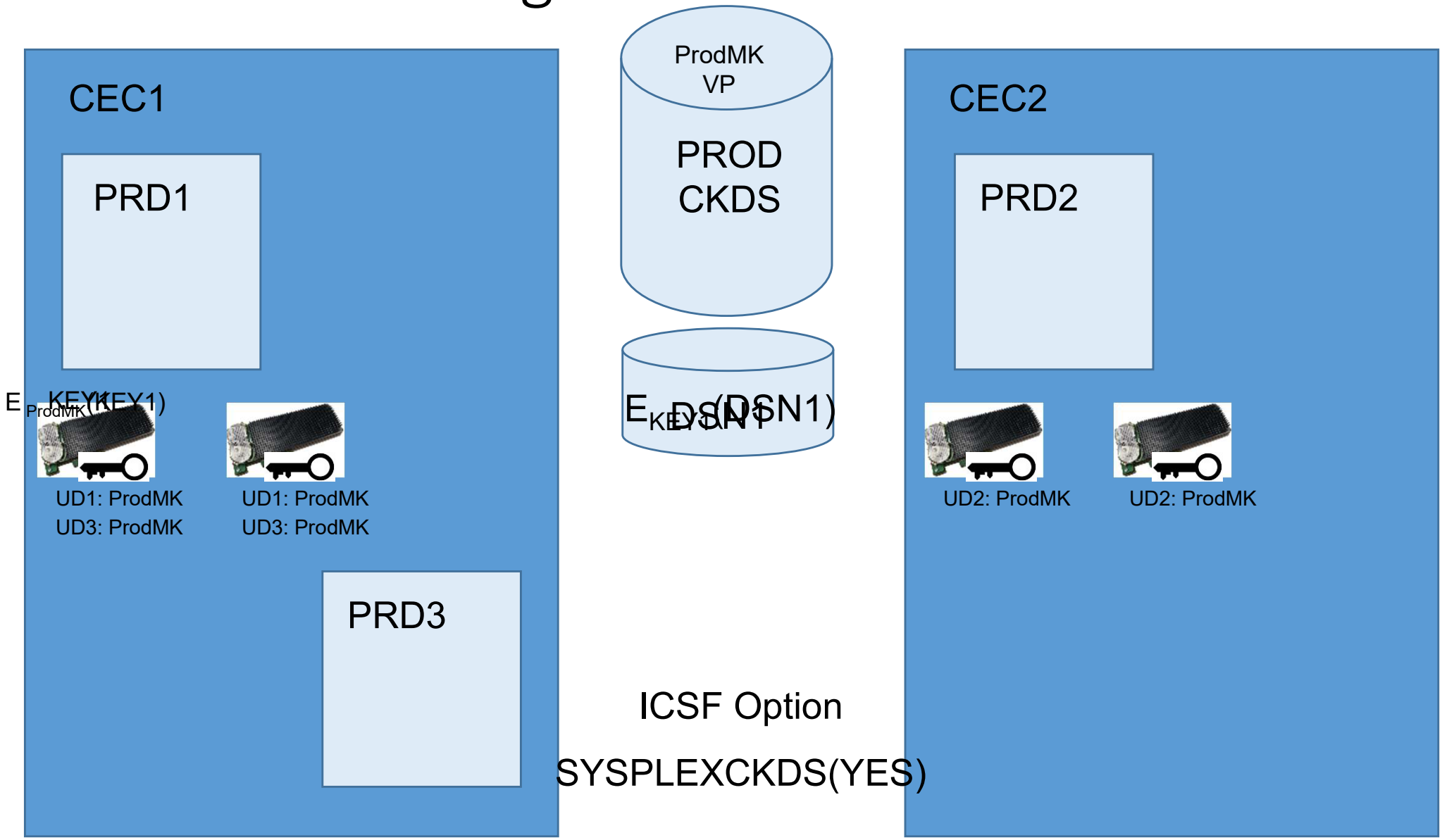
February 2022

Copyrights and Trademarks

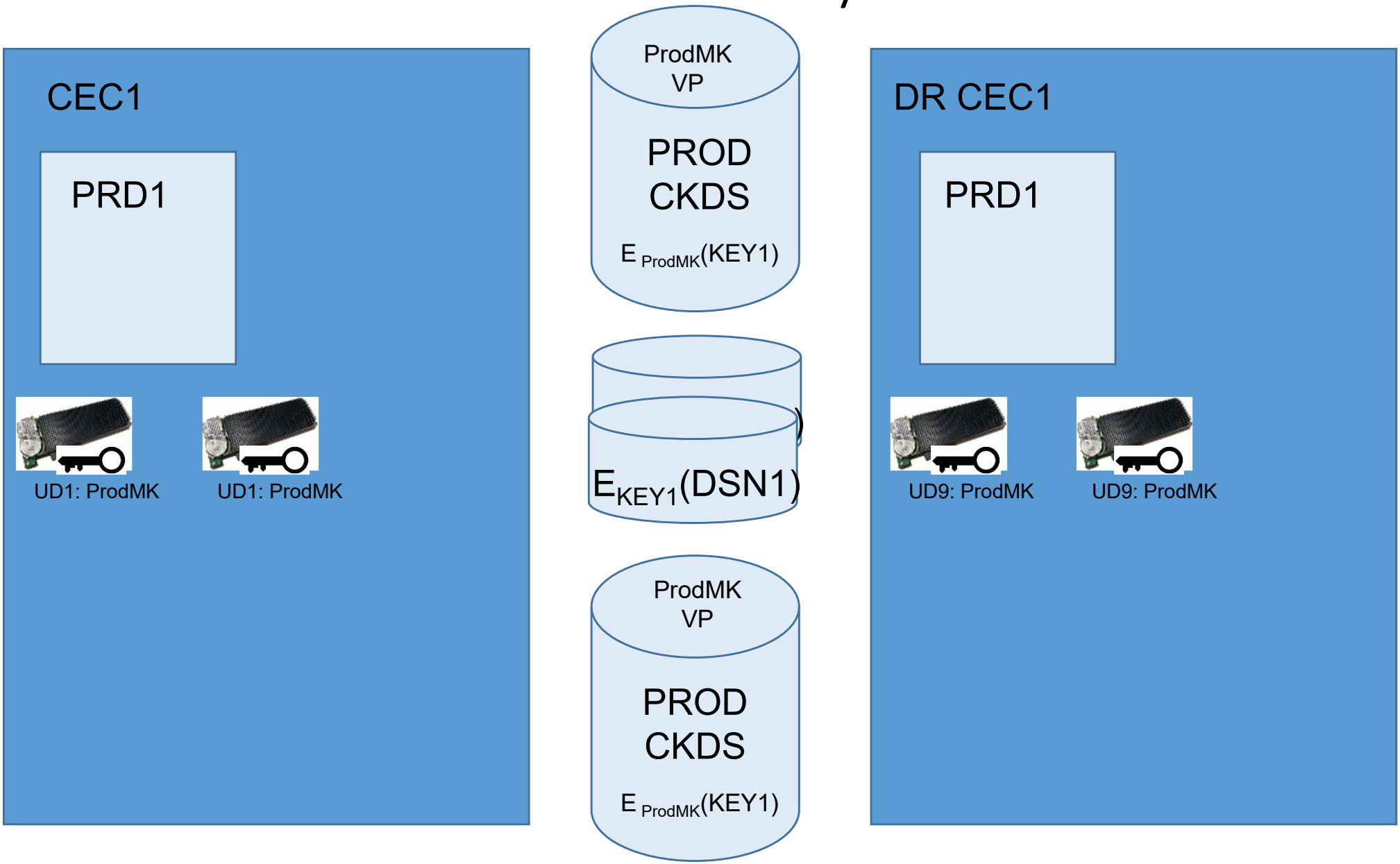
- Copyright © 2022 Greg Boyd, Mainframe Crypto, LLC. All rights reserved.
- Presentation based on material copyrighted by IBM, and developed by myself, as well as many others that I worked with over the past 30+ years

- All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. IBM, System z, zEnterprise and z/OS are trademarks of International Business Machines Corporation in the United States, other countries, or both. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.
- **THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY**. Greg Boyd and Mainframe Crypto, LLC assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will Greg Boyd or Mainframe Crypto, LLC be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if expressly advised in advance of the possibility of such damages.

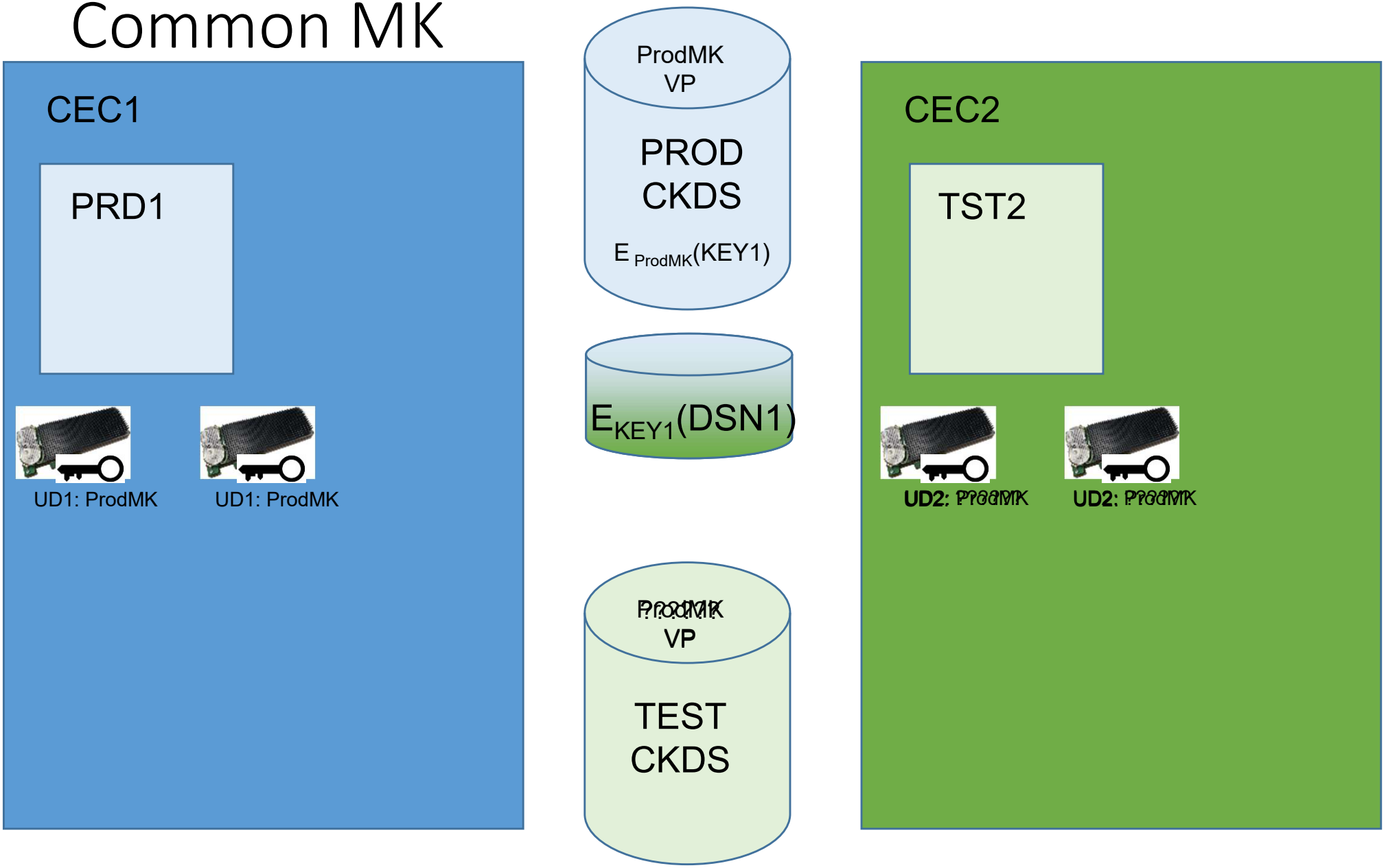
CSFPlex Sharing Data Sets



Also how Disaster Recovery Works



Shared Data, Unique key store, Common MK



Common master key - Copy the key

- IDCAMS REPRO
- KeyXfer Tool (GitHub)

IDCAMS REPRO

```
//SAVEKEYS EXEC PGM=IDCAMS,REGION=4M
//CKDSP    DD DSN=PROD.CSF.CSFCKDS,DISP=SHR          Primary PROD CKDS
//SEQP     DD DSN=savekey.dsn,DISP=(,CATLG),          Bkup key data set
//              SPACE=(TRK,(1,1)),UNIT=SYSALLDA,
//              DCB=(LRECL=372,BLKSIZE=2048,RECFM=VB)
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
   REPRO INFILE(CKDSP) OUTFILE(SEQP) –
         FROMKEY(key label to be copied) –
         COUNT(1)
```

IDCAMS REPRO

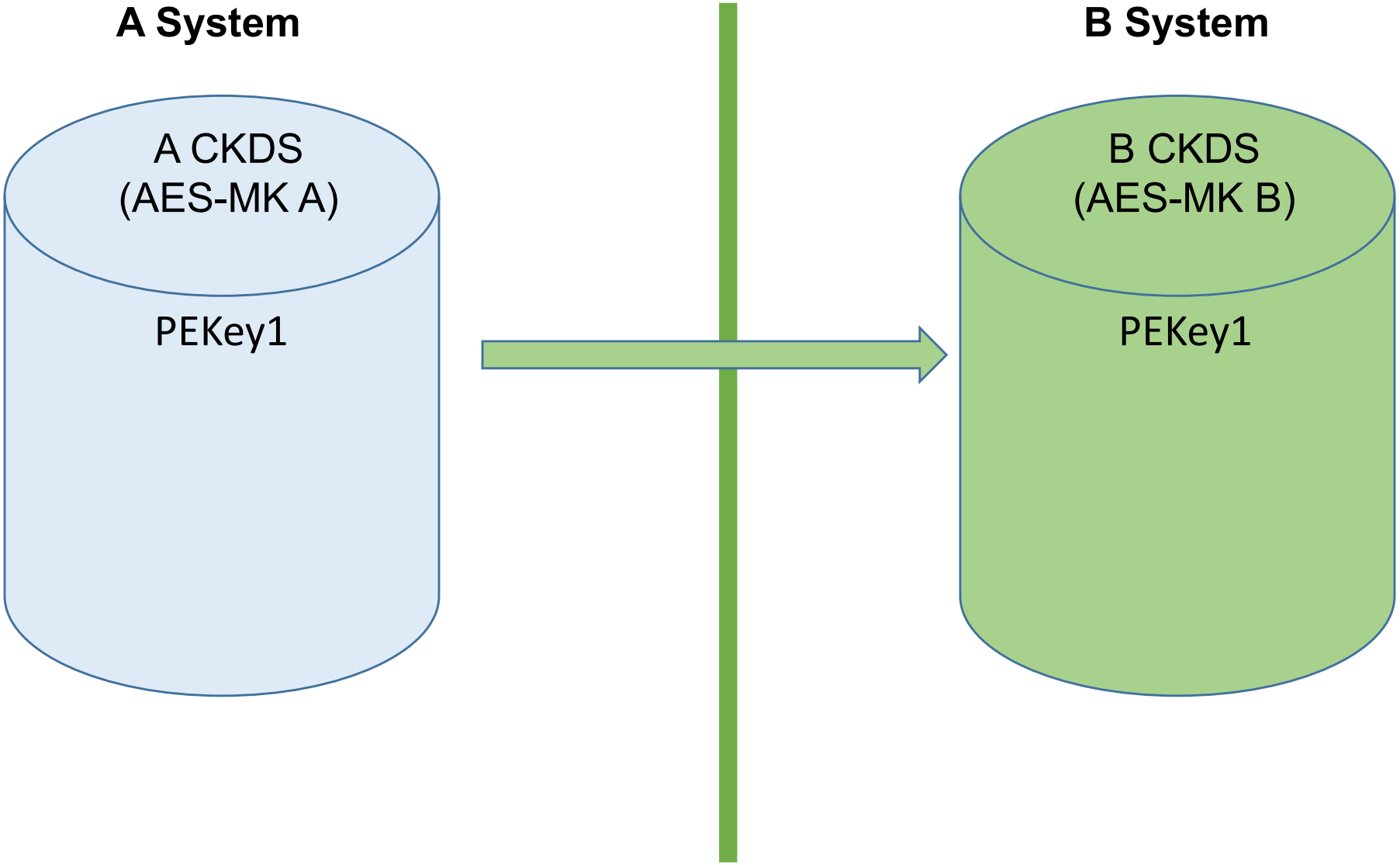
```
//RESTKEY EXEC PGM=IDCAMS,REGION=4M
//SEQT       DD  DSN=savekey.dsn,DISP=SHR                    Bkup key data set
//CKDST      DD  DSN=TEST.CSF.CSFCKDS,DISP=SHR               Primary CKDS
//SYSPRINT DD SYSOUT=*
//SYSIN      DD *
  REPRO INFILE(SEQT) OUTFILE(CKDST)
/*
```

KeyXfer Tool – REXX EXEC on GitHub

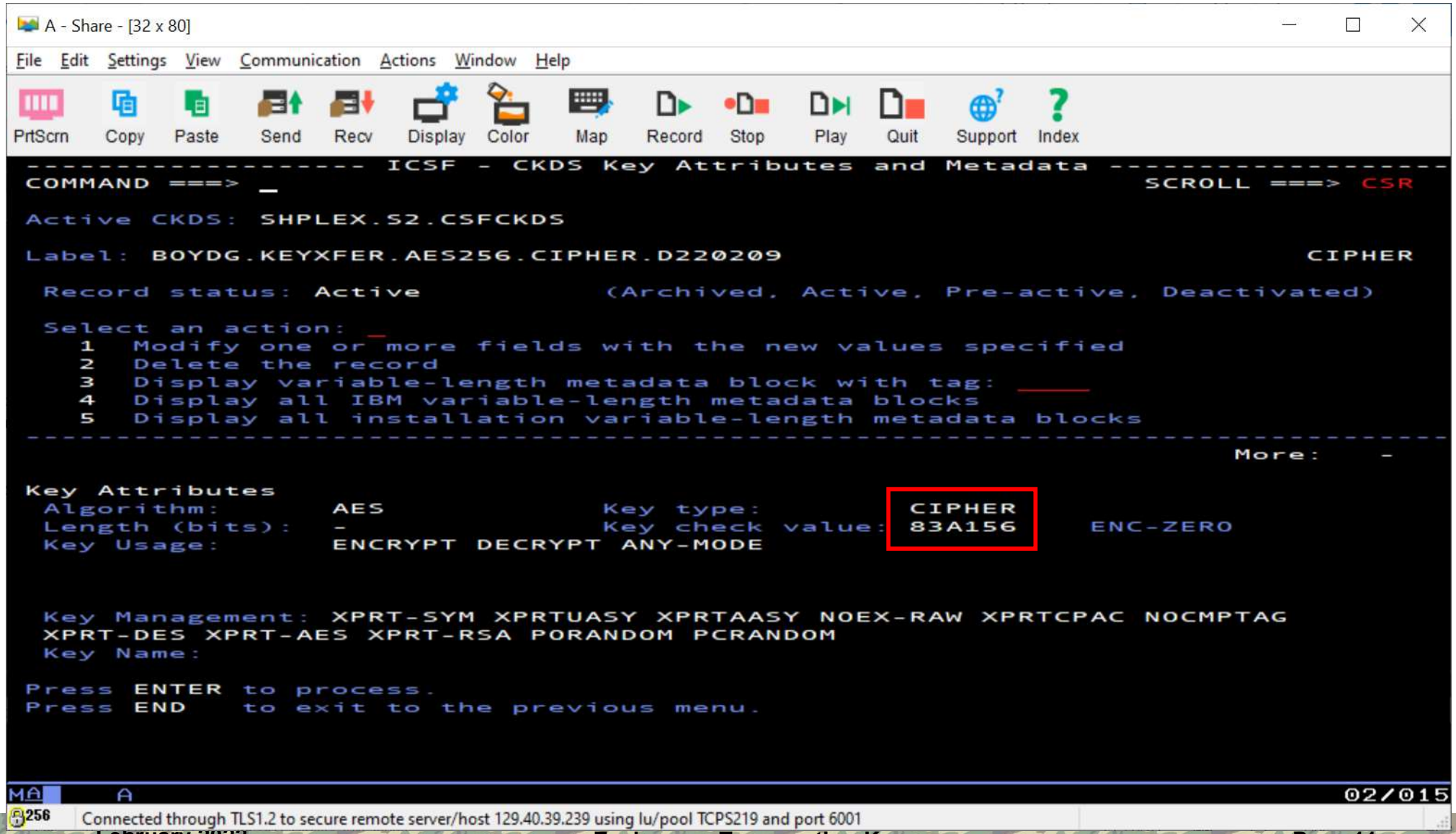
- KEYXFER OPER, LABEL, DSN, OPTION
  - OPER
    - WRITE_CKDS (read the key from the CKDS; write to file)
    - READ_CKDS (read from file; write to CKDS)
    - WRITE_PKDS (read the key from the PKDS; write to file)
    - READ_PKDS (read from file; write to PKDS)
  - LABEL – fully qualified key label
  - DSN – PS or PDS(member) to store the key for transfer

    > 01/23/22 6:35PM
    BOYDG.PE.KEY.20211221
    010000000400C082 … first 32 bytes of the key token
    C220FBC59BF … last 32 bytes of the key token
  - OPTION
    - OVERWRITE
    - DEBUG

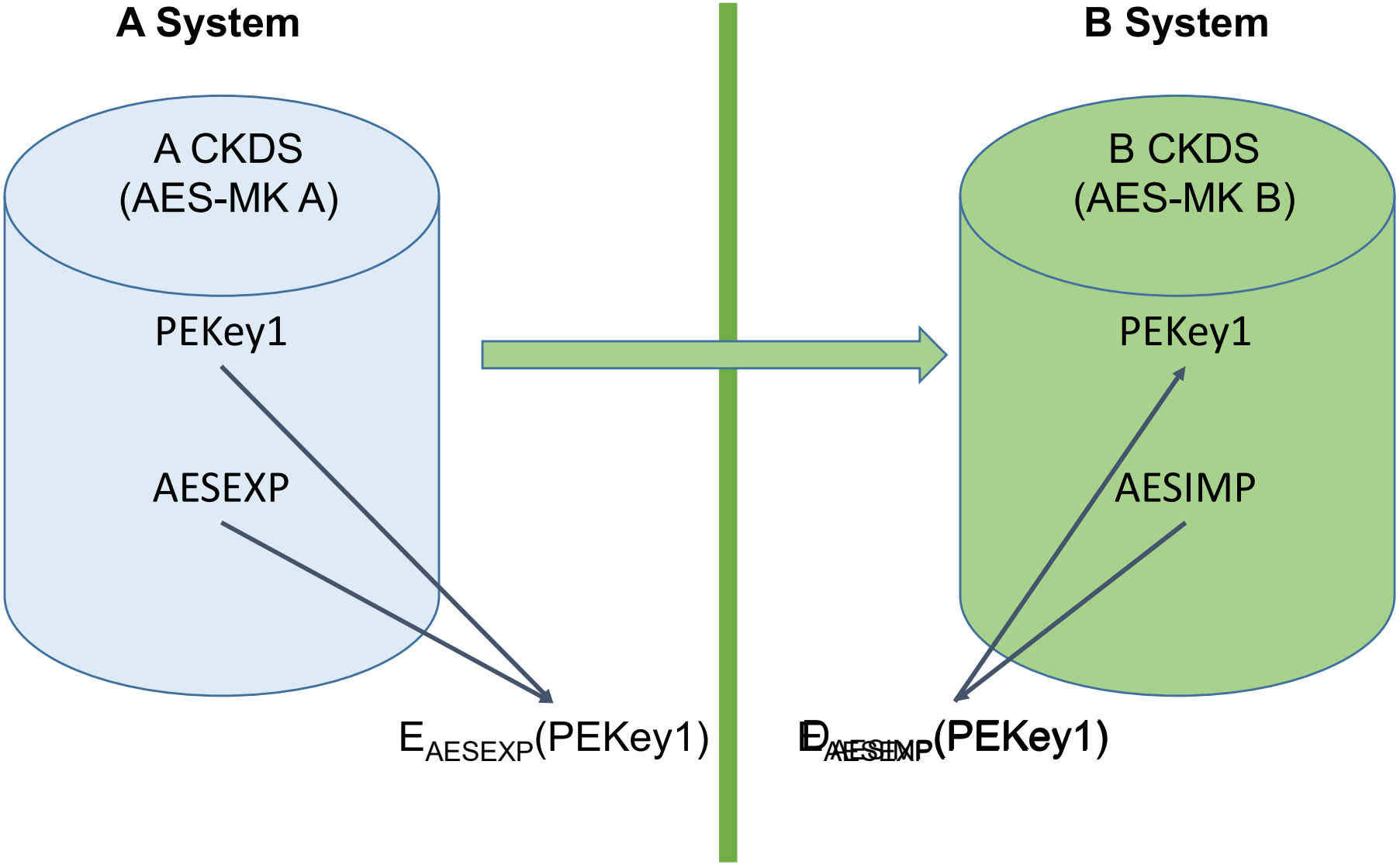
Moving keys between systems



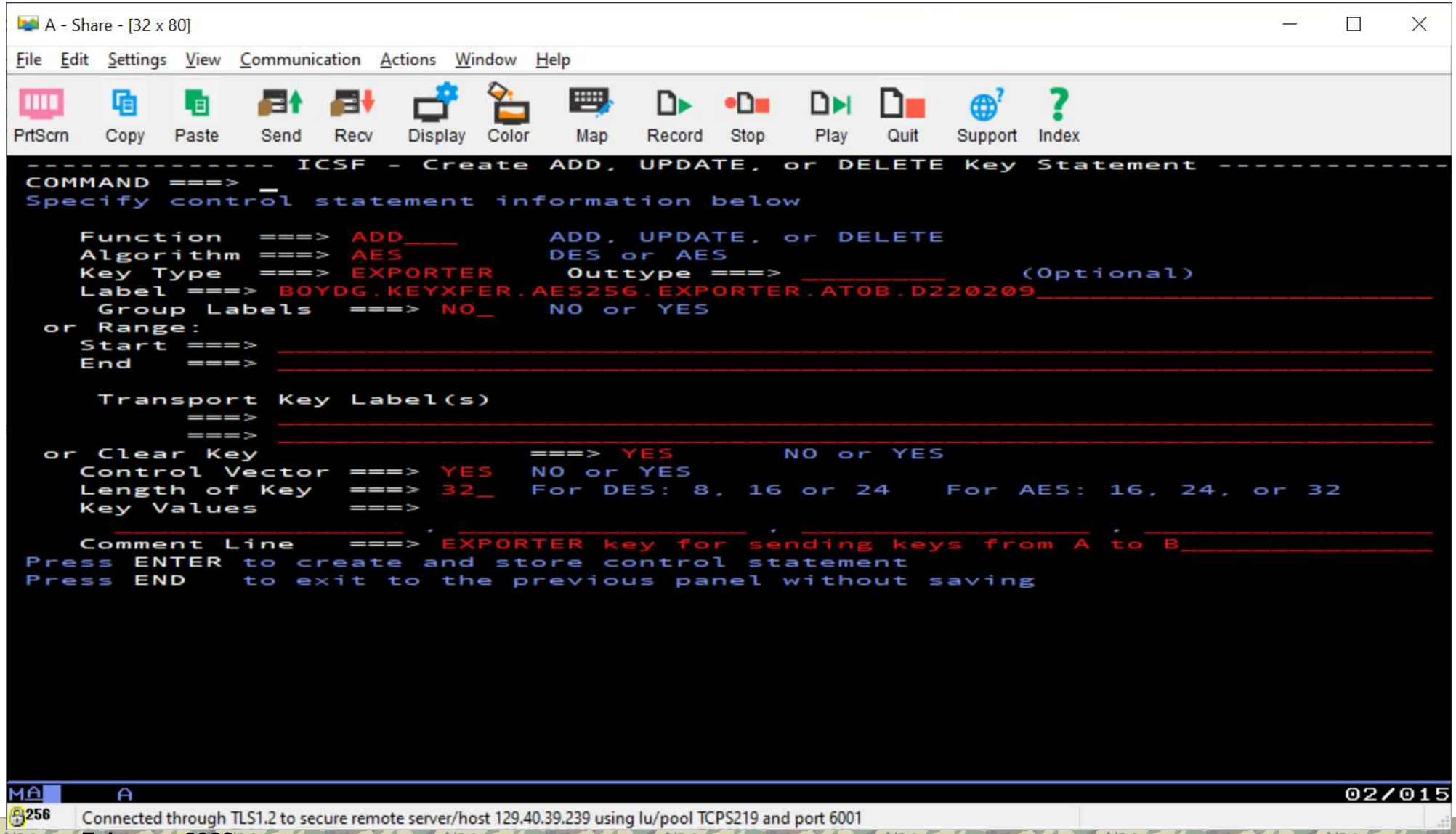
PEKEY1 – CIPHER key with KCV 83A156



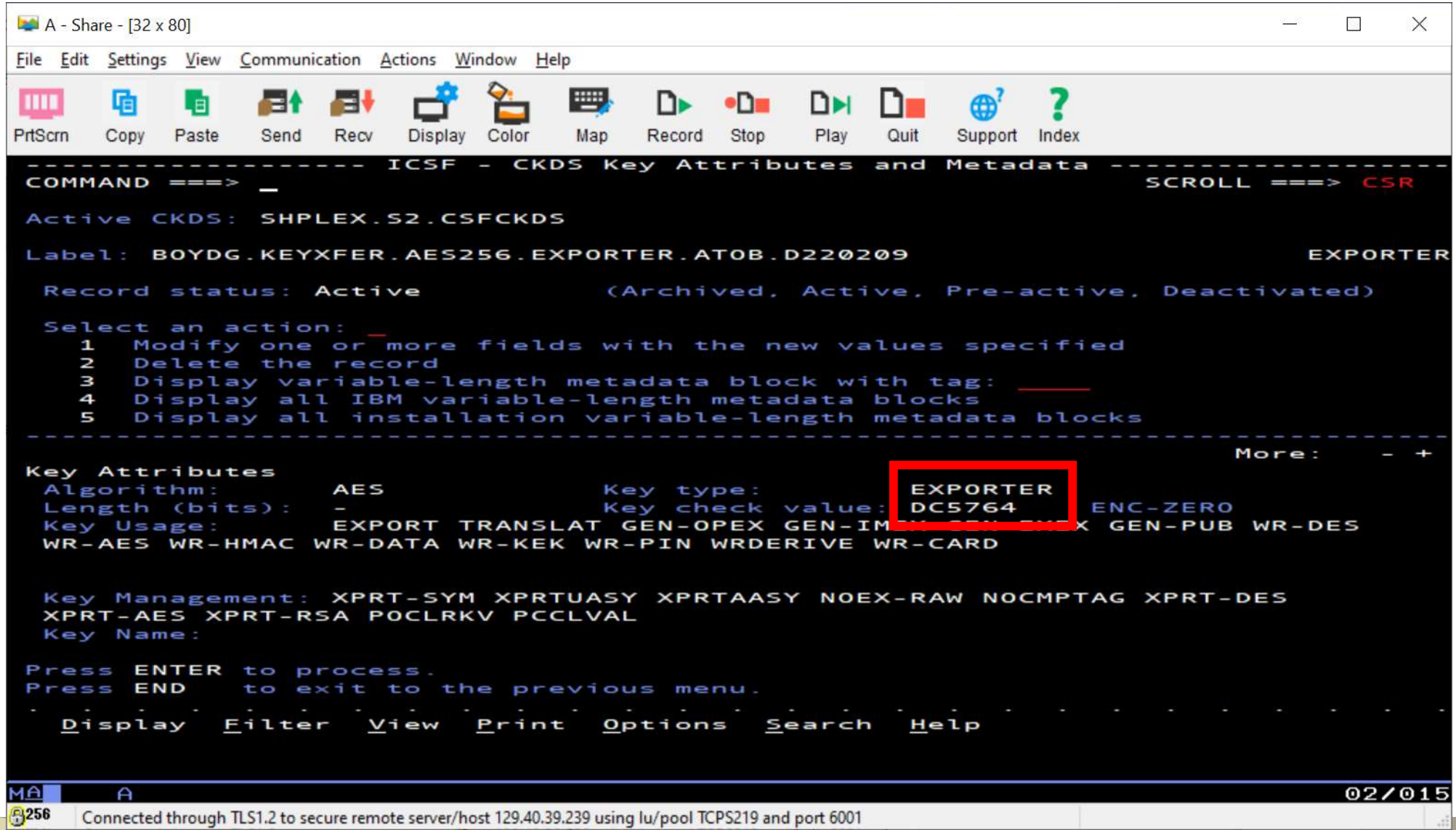
Moving keys between systems



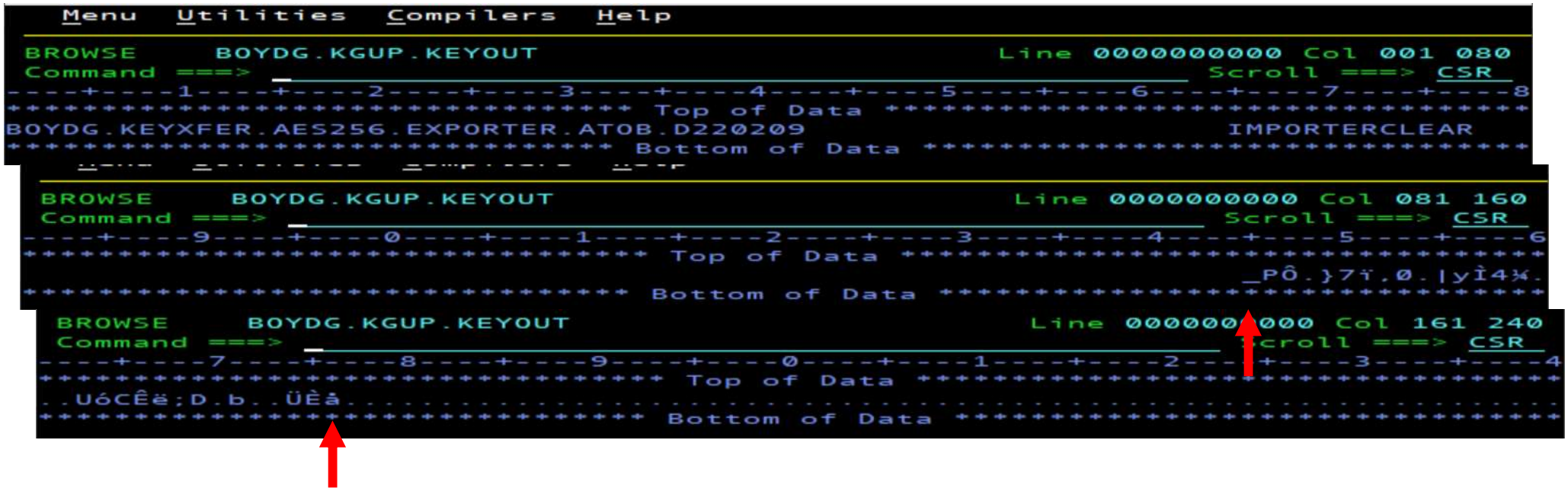
```
/* EXPORTER KEY FOR SENDING KEYS FROM A TO B */
ADD TYPE(EXPORTER) LENGTH(32) ALGORITHM(AES),
 CLEAR,
 LAB(BOYDG.KEYXFER.AES256.EXPORTER.ATOB.D220209)
```



CKDS Keys Utility View of Exporter Key



KEYOUT Data Set

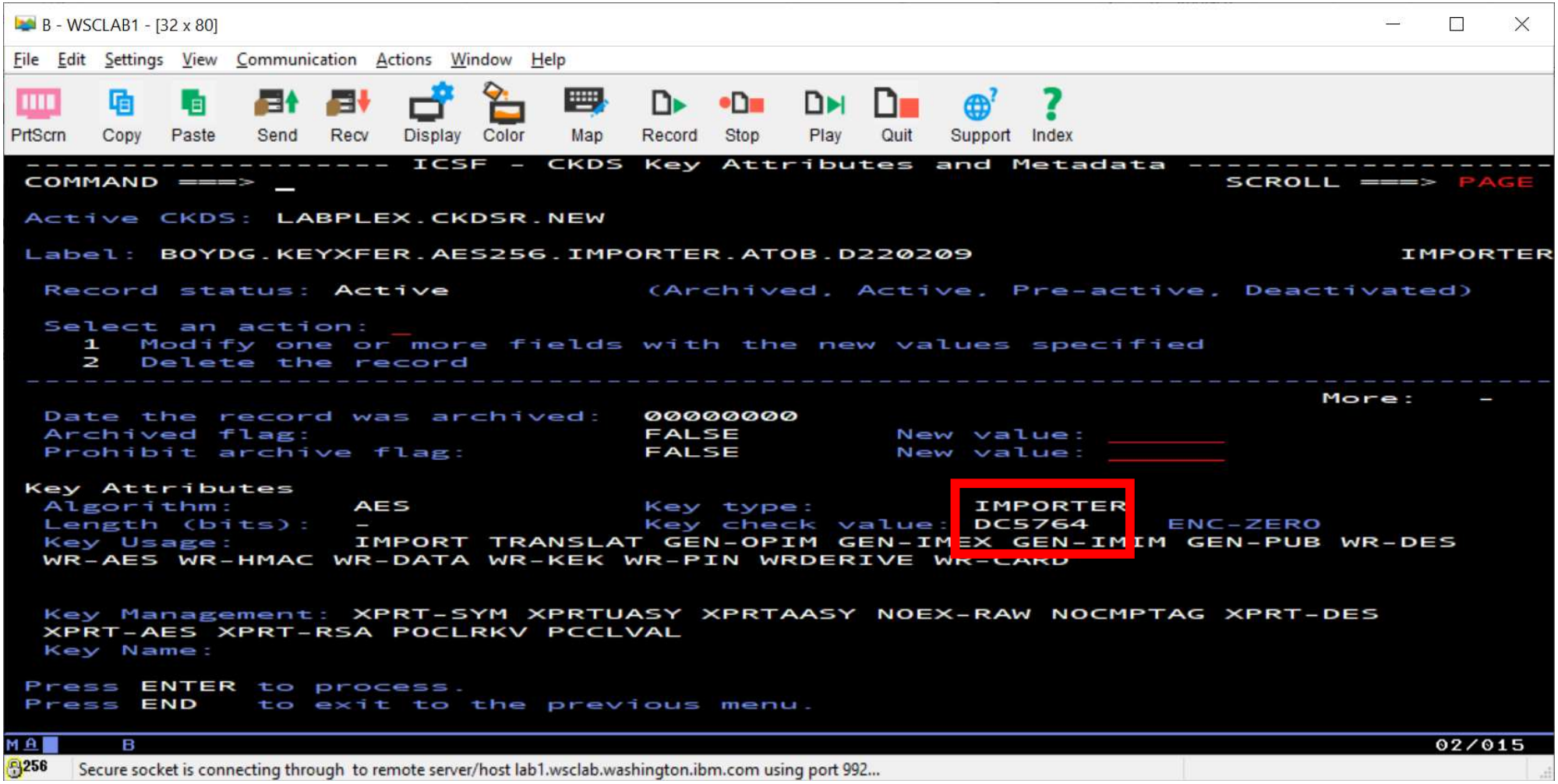


- 64 Bytes — Key Label
- 8 Bytes — Key Type
- 64 Bytes — Either the TRANSKEY label or CLEAR
- 8 Bytes — Key Type
- Variable Length — Key Token

Control Statements for Target system

ADD LABEL(BOYDG.KEYXFER.AES256.IMPORTER.ATOB.D220209) ,
TYPE(IMPORTER) CLEAR ,
ALGORITHM(AES) ,
KEY(6DD7EB29D0F7576B, 80094FA878F4B71A, 2D35E4CEC372535E,
C43A823337FC7447)

New Key Label, New Key Type, Same KCV



Transferring the operational key

- CSNBKEX – Key Export
  - Takes a key from being encrypted under a master key to being encrypted under an exporter key-encrypting key
- CSNBKIM – Key Import
  - Takes a key from being encrypted under an importer key-encrypting key to being encrypted under a master key
- CSNDSYX
  - Takes a symmetric key from being encrypted under a master key to being encrypted under an RSA public key or AES Exporter key
- CSNDSYI
  - Takes a symmetric AES or DES DATA key from being encrypted under an RSA public key to being encrypted under a master key
- CSNDSYI2
  - Takes a symmetric HMAC, AES or DES key from being encrypted under an RSA public key or an AES EXPORTER key to being encrypted under a master key

Transporting AES keys

- Multi-step process to wrap/unwrap an existing AES DATA key with AES IMPORTER/EXPORTER keys derived from an ECC Key Pair. https://community.ibm.com/community/user/ibmz-and-linuxone/viewdocument/drvaesmp?CommunityKey=6593e27b-caf6-4f6c-a8a8-10b62a02509c&tab=librarydocuments

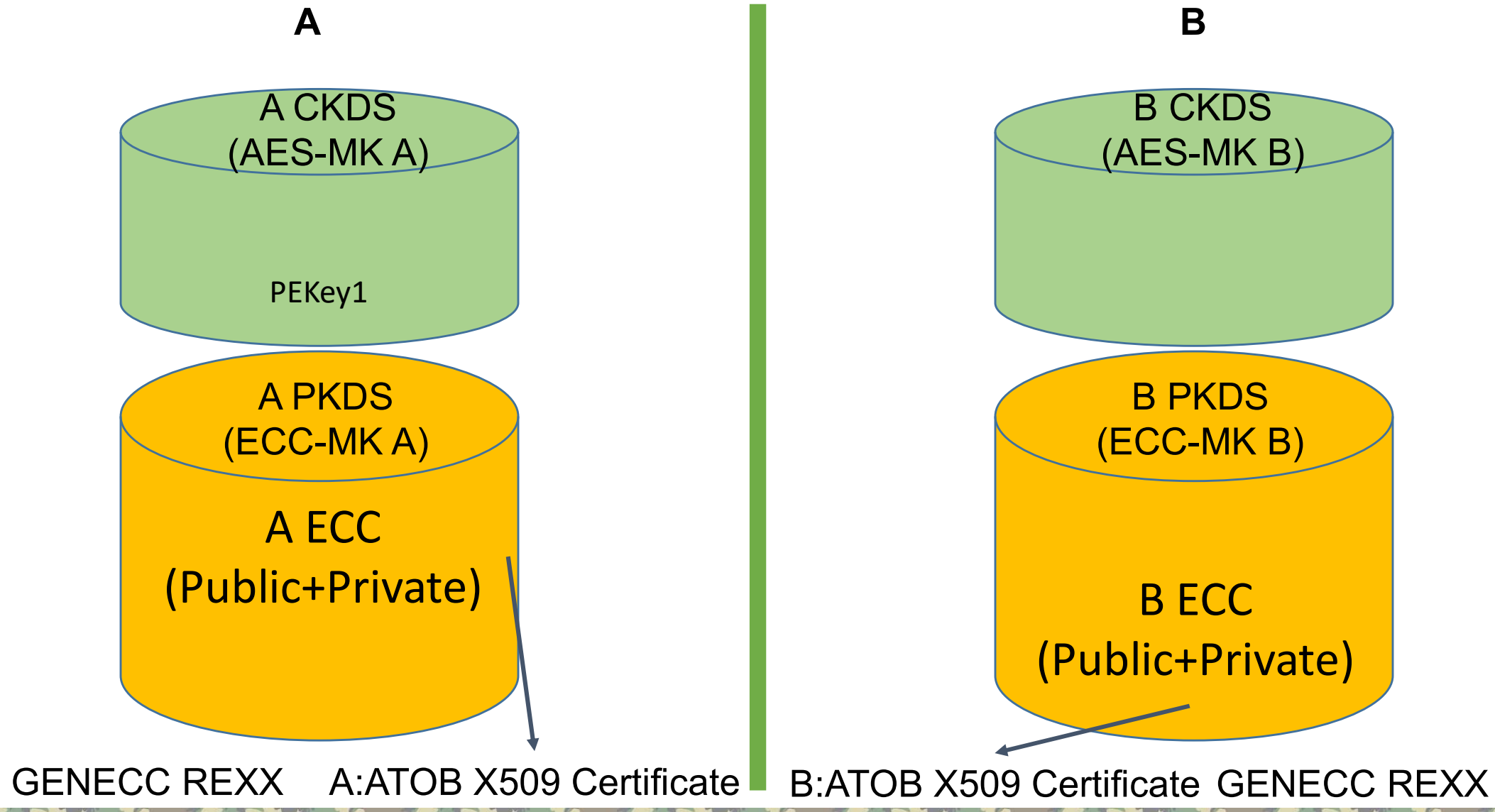
- Set of REXX EXECs to
  - GENECC2 - Create a public/private key pair on A and export the public key to B
  - GENECC2 - Create a public/private key pair on B and export the public key to A
  - IMPRTEC2 – On A, import B's public key
  - IMPRTEC2 – On B, import A's public key
  - DRVAESXP - On A, create an EXPORTER key using A's private key and B's public key
  - DRVAESMP - On B, create an IMPORTER key using B's private key and A's public key
  - ESPAES32 - On A, export the operational key using the EXPORTER key
  - IMPAES32 - On B, import the operational key using the IMPORTER key

Diffie-Hellman Key Exchange

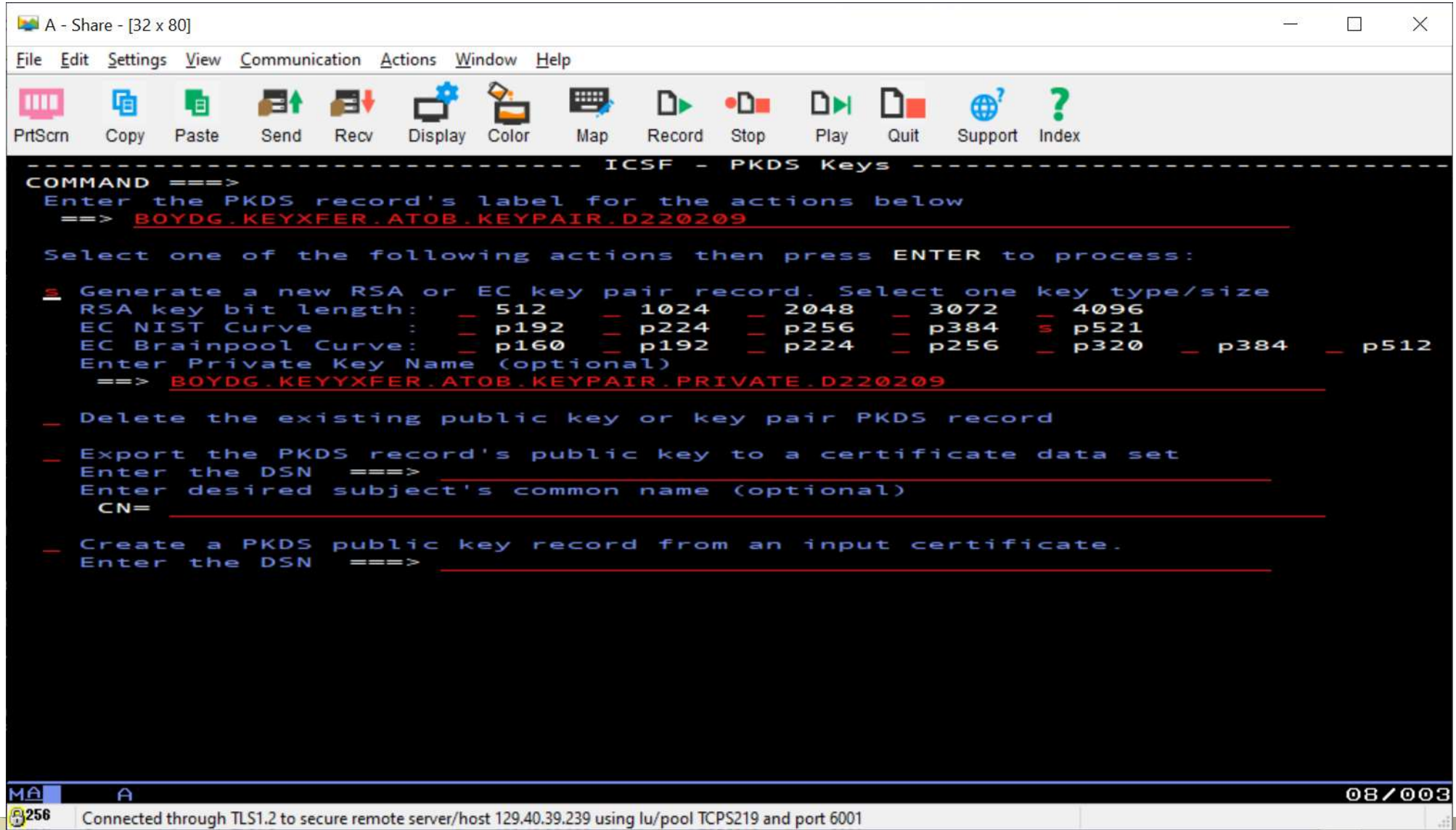
- Alice and Bob agree to use specific values for a modulus and base
  - Modulus p 23
  - Base g 5
- Alice selects a secret value … we'll use 4 and
  - Calculates $A = g^a \bmod p \Rightarrow A = 5^4 \bmod 23 = 4$
  - And shares A with Bob
- Bob selects a secret value … we'll use 3 and
  - Calculates $B = g^b \bmod p \Rightarrow B = 5^3 \bmod 23 = 10$
  - And shares B with Alice
- Alice computes the shared secret s
  - $s = B^a \bmod p \Rightarrow 10^4 \bmod 23 = 18$
- Bob computes their shared secret s
  - $S = A^b \bmod p \Rightarrow 4^3 \bmod 23 = 18$

From Wikipedia:
https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

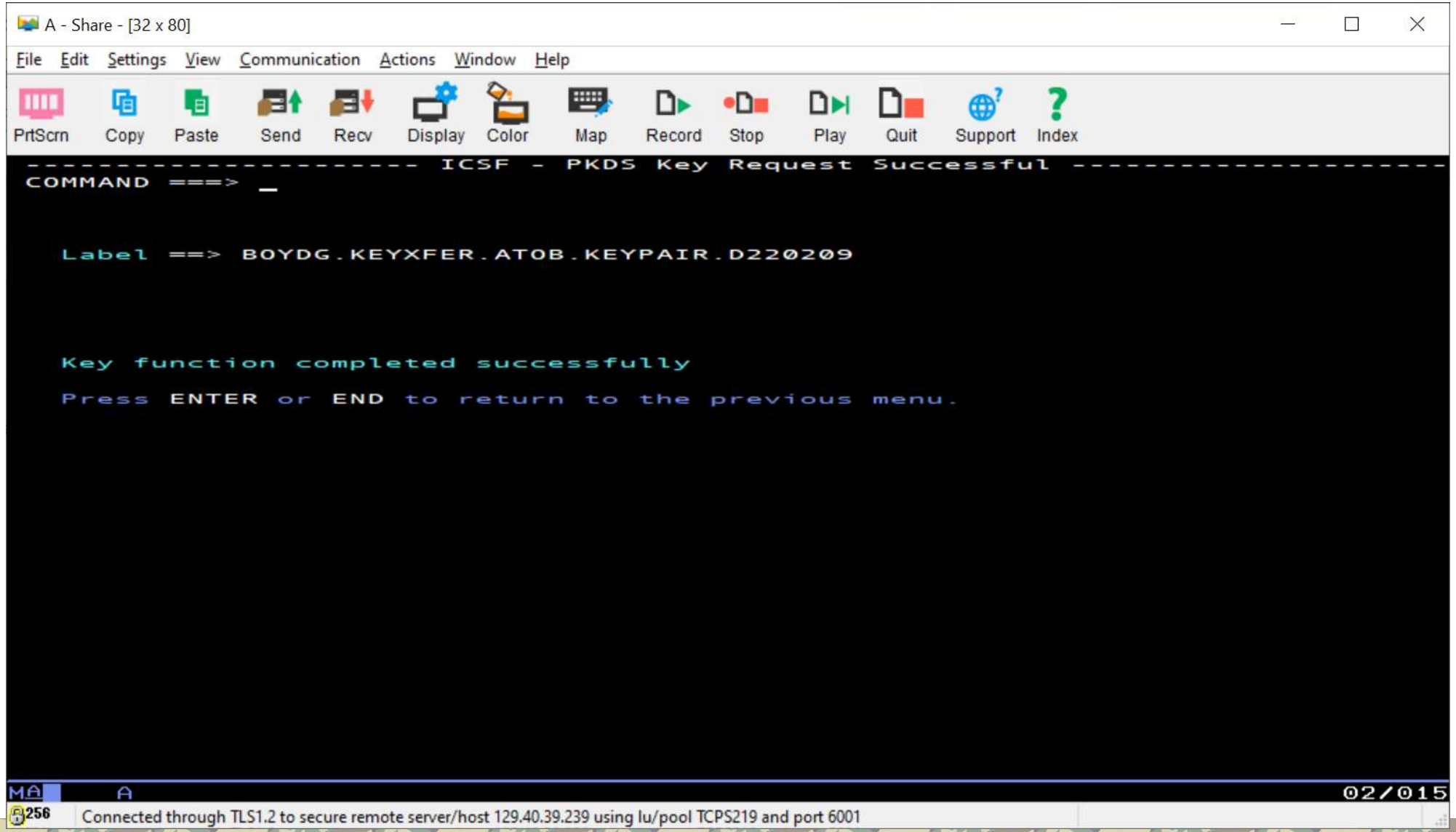
Create ECC Pair for AtoB and BtoA



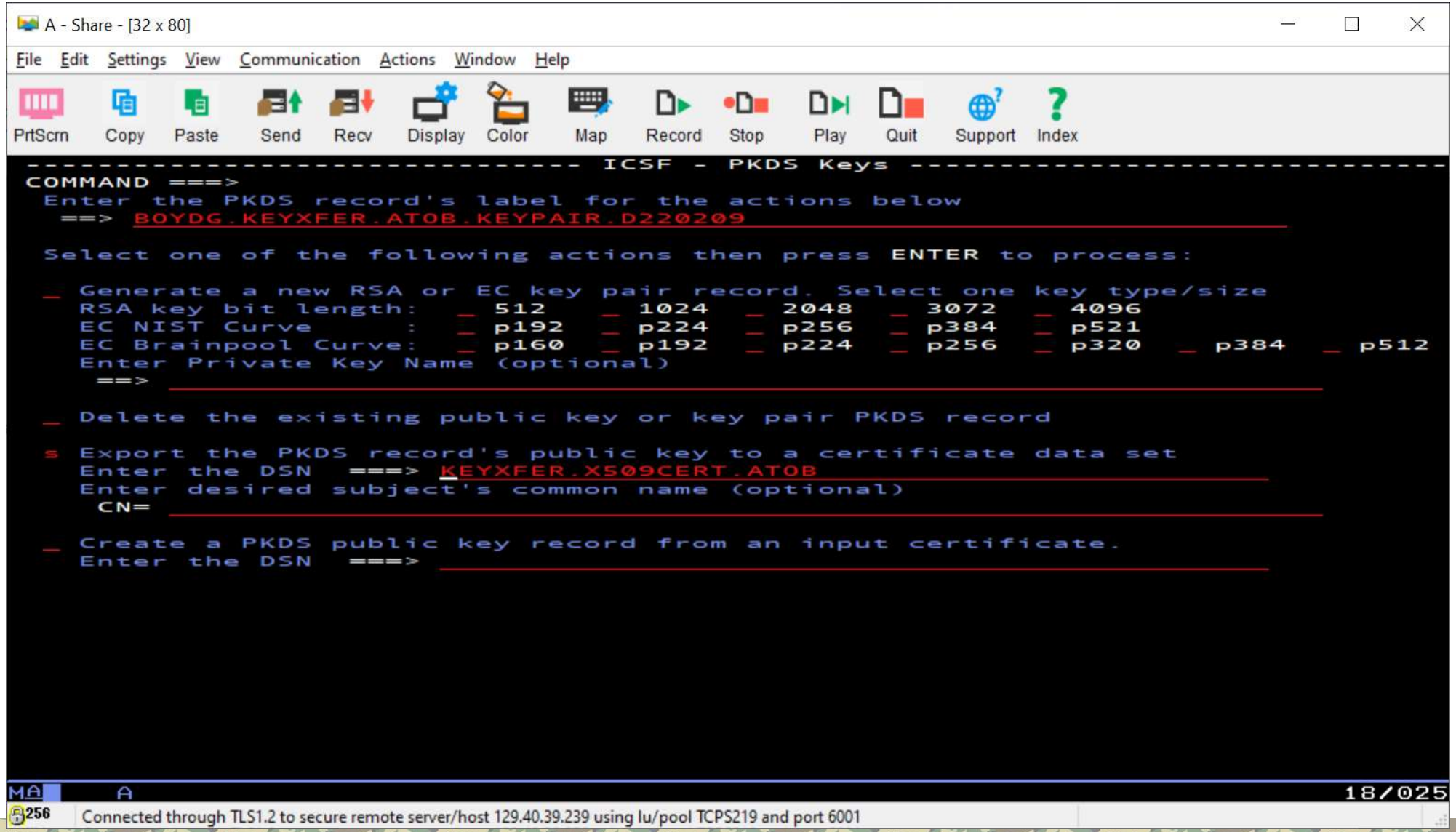
On A-Generate ECC Public/Private Key Pair



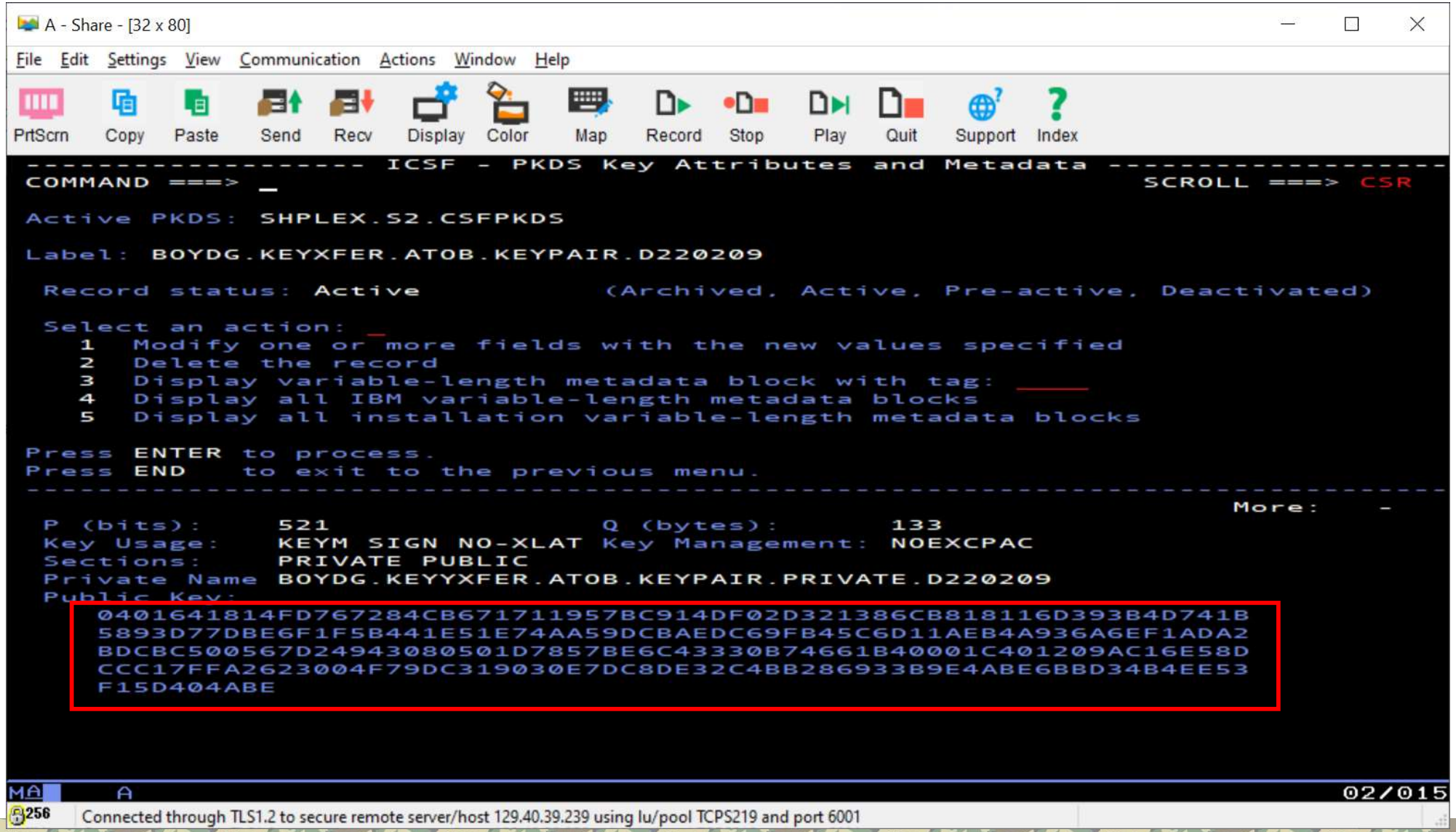
On A - Key Pair Generated Successfully



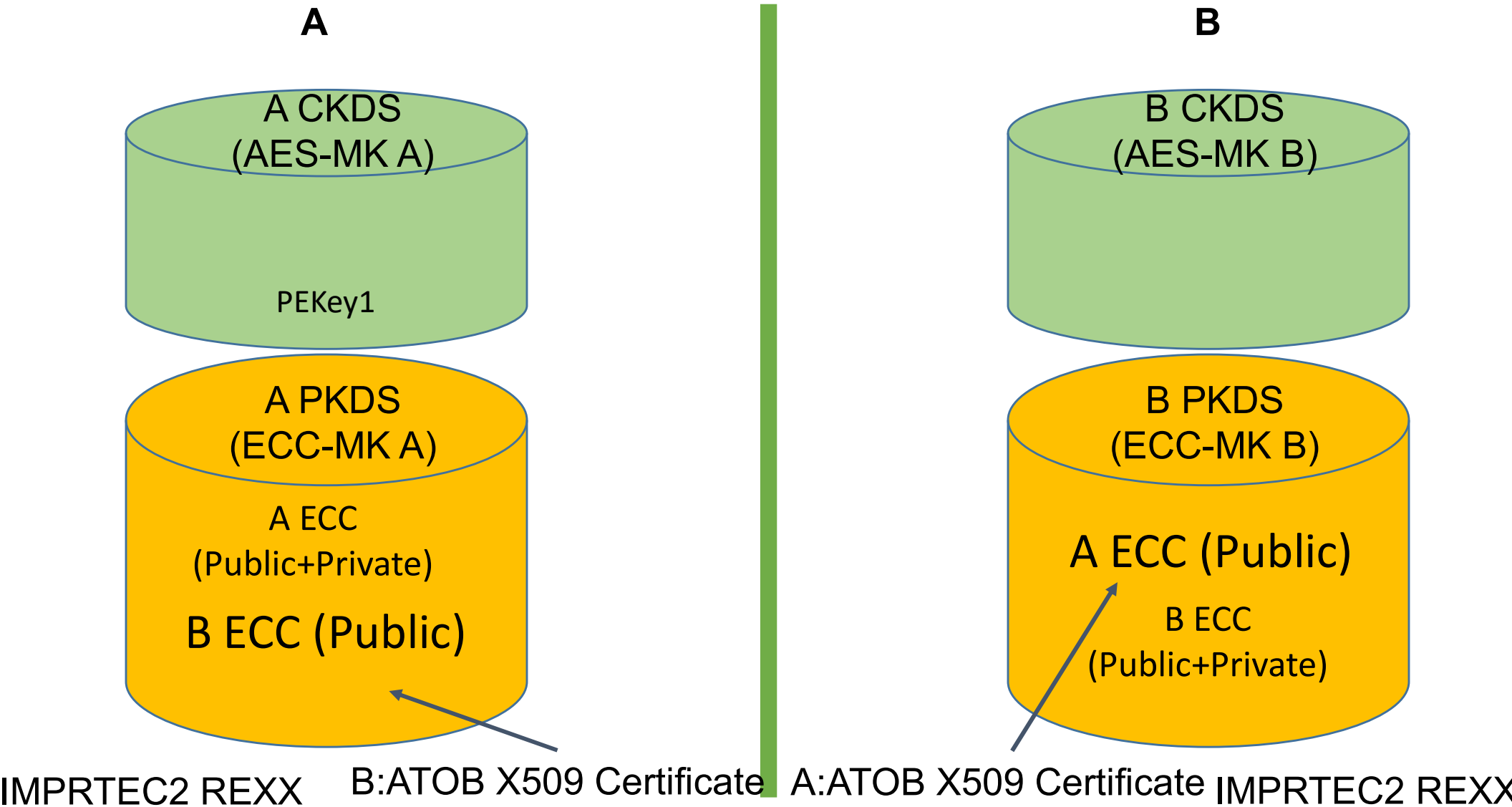
On A - Export the public key



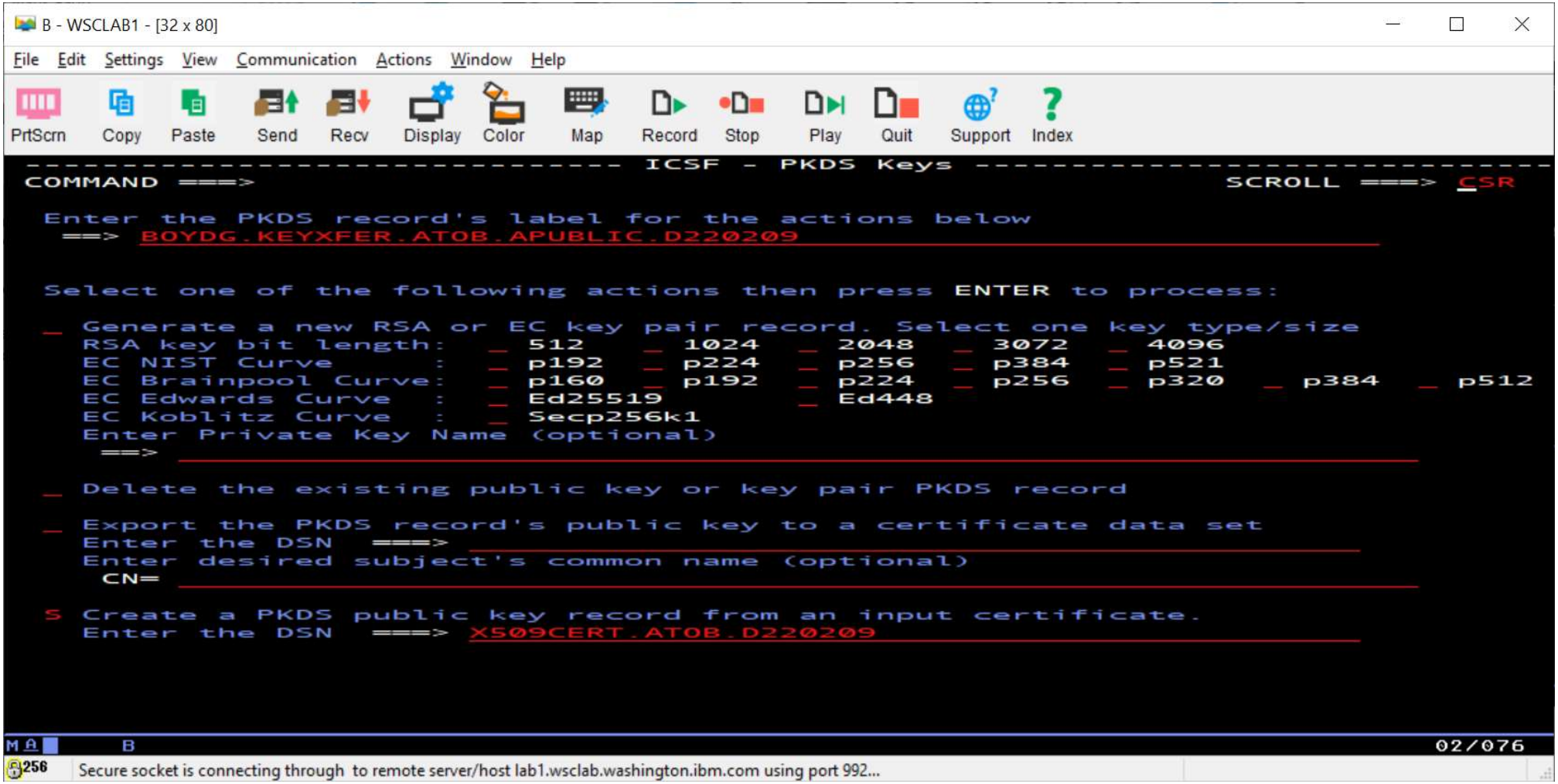
On A – PKDS Keys Utility



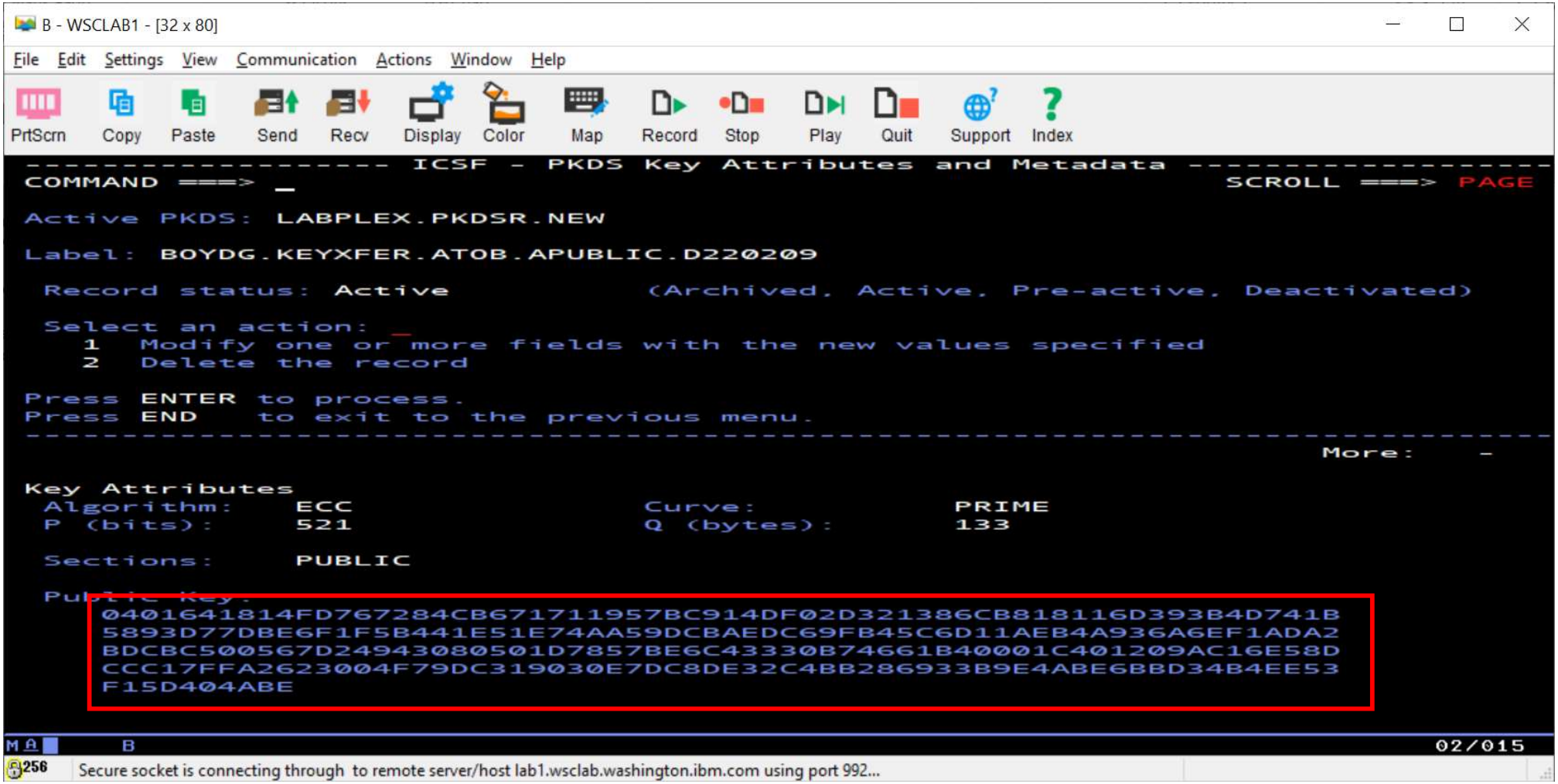
Import A's Public Key on B



On B - Import the Public key from A



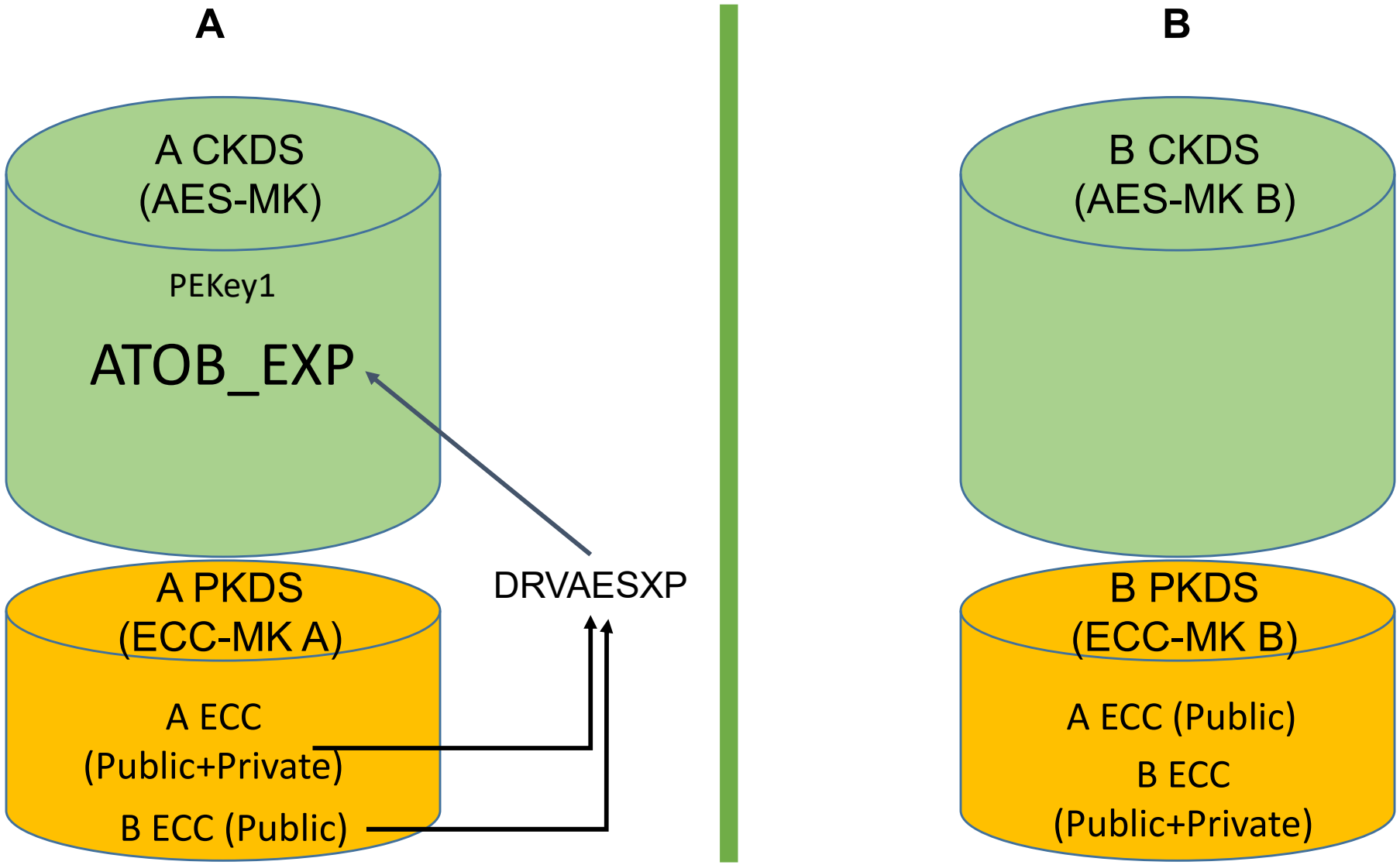
On B – A's Public key is available



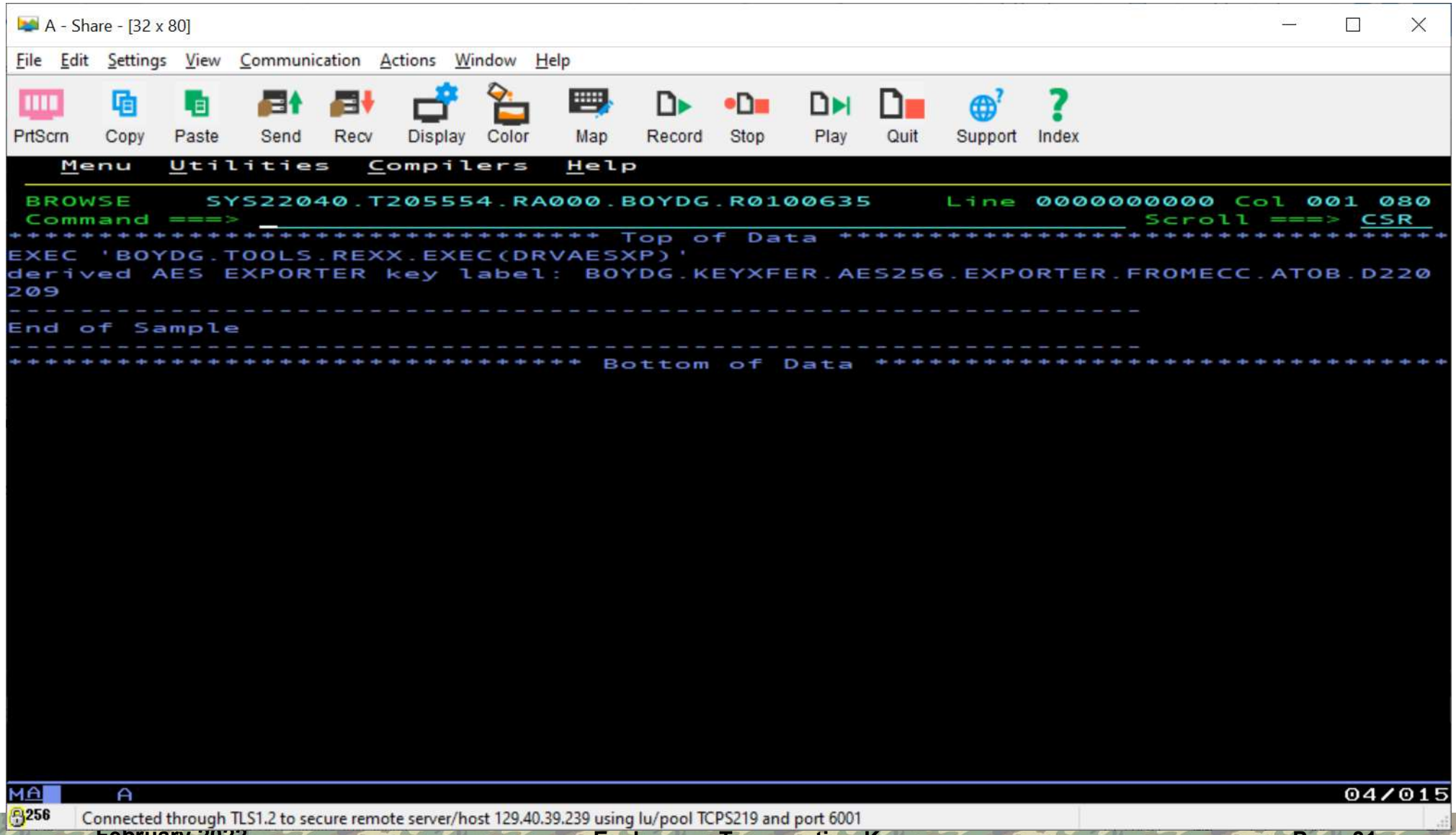
Repeat for B to A

- Generate a public/private key pair on B
- Export the public key from B to a file
- Move that certificate file to A
- Import the public key from B, on A

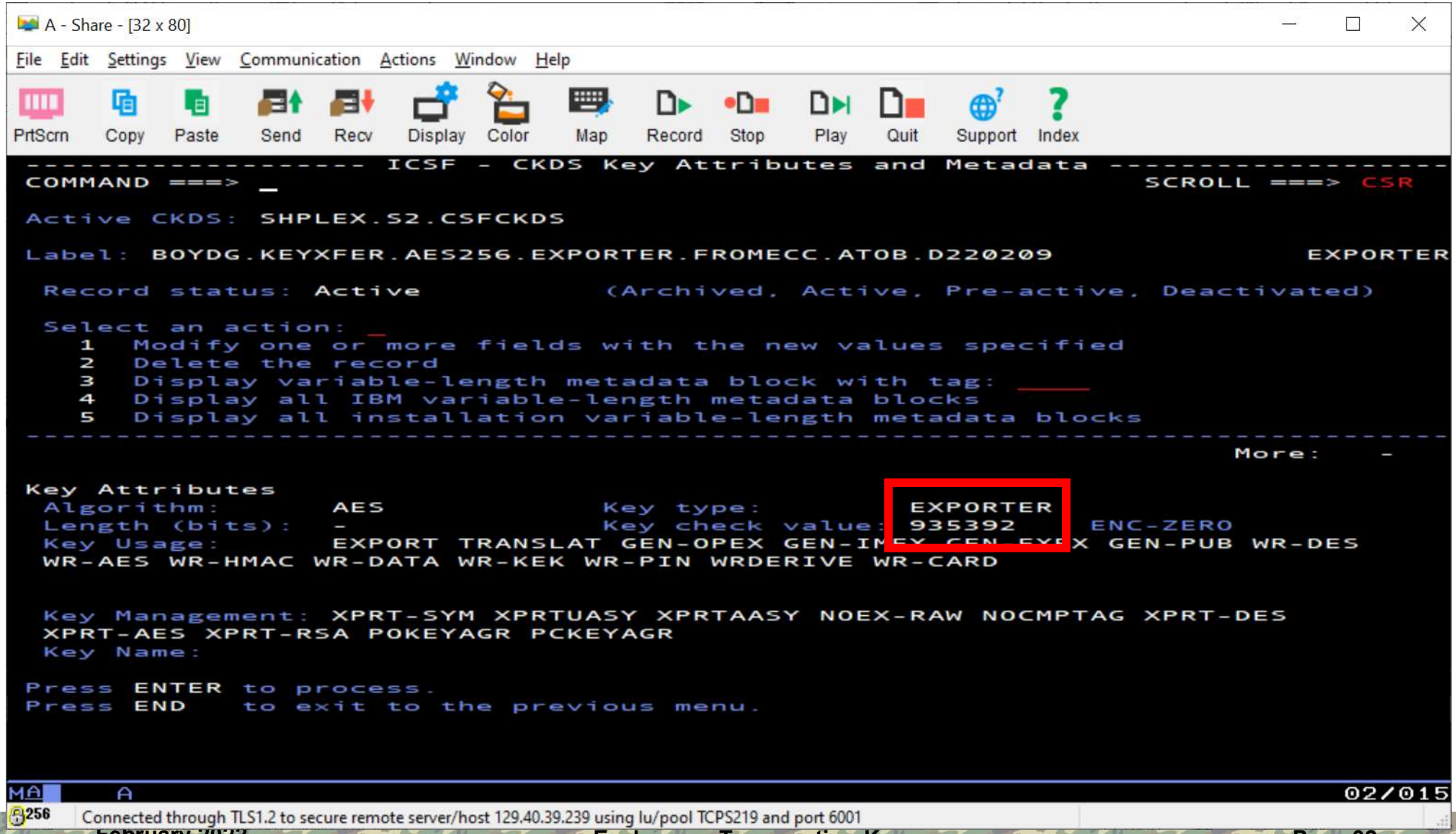
Create AES Exporter on A



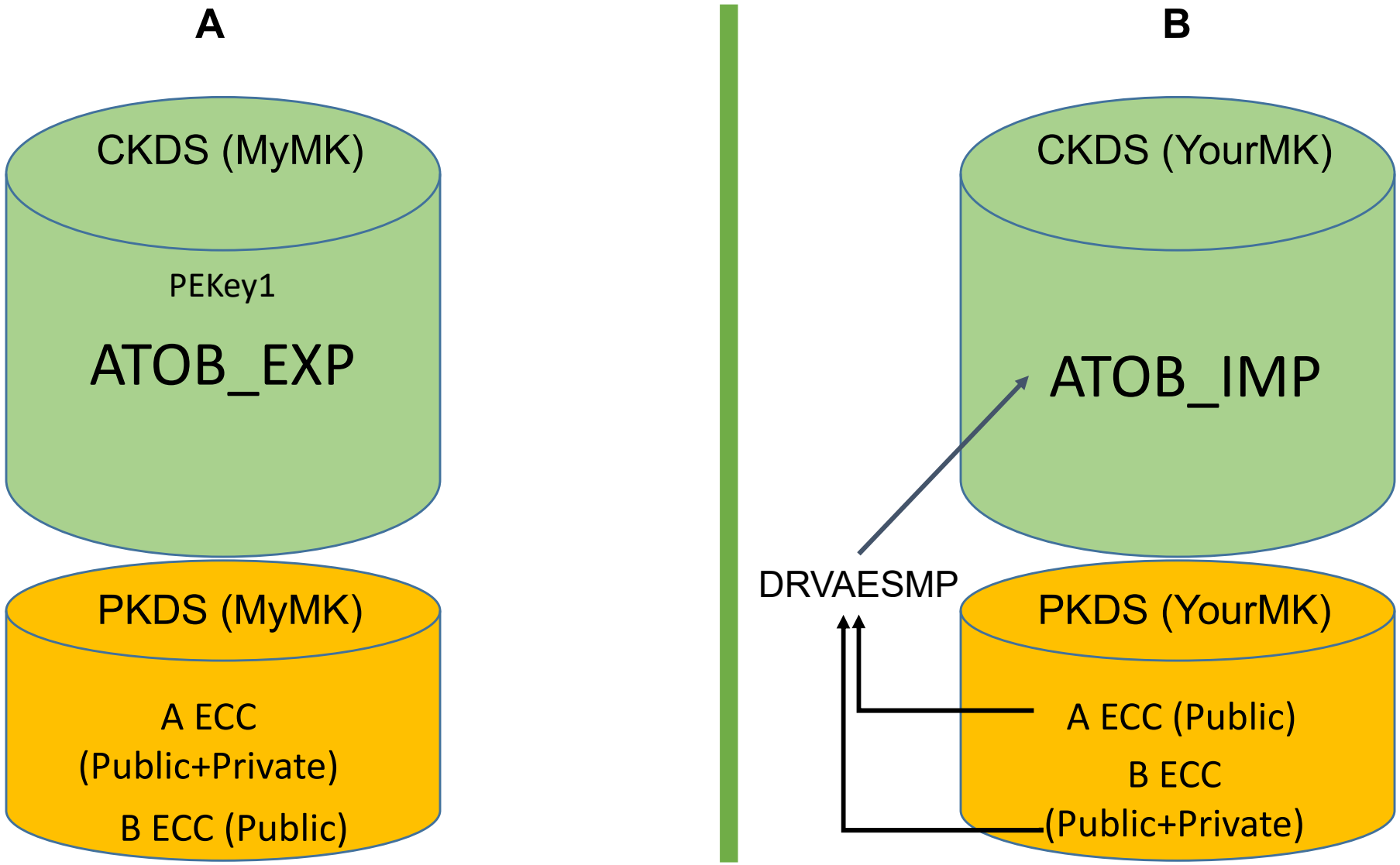
On A – Execute DRVAESXP REXX to Create the Exporter key



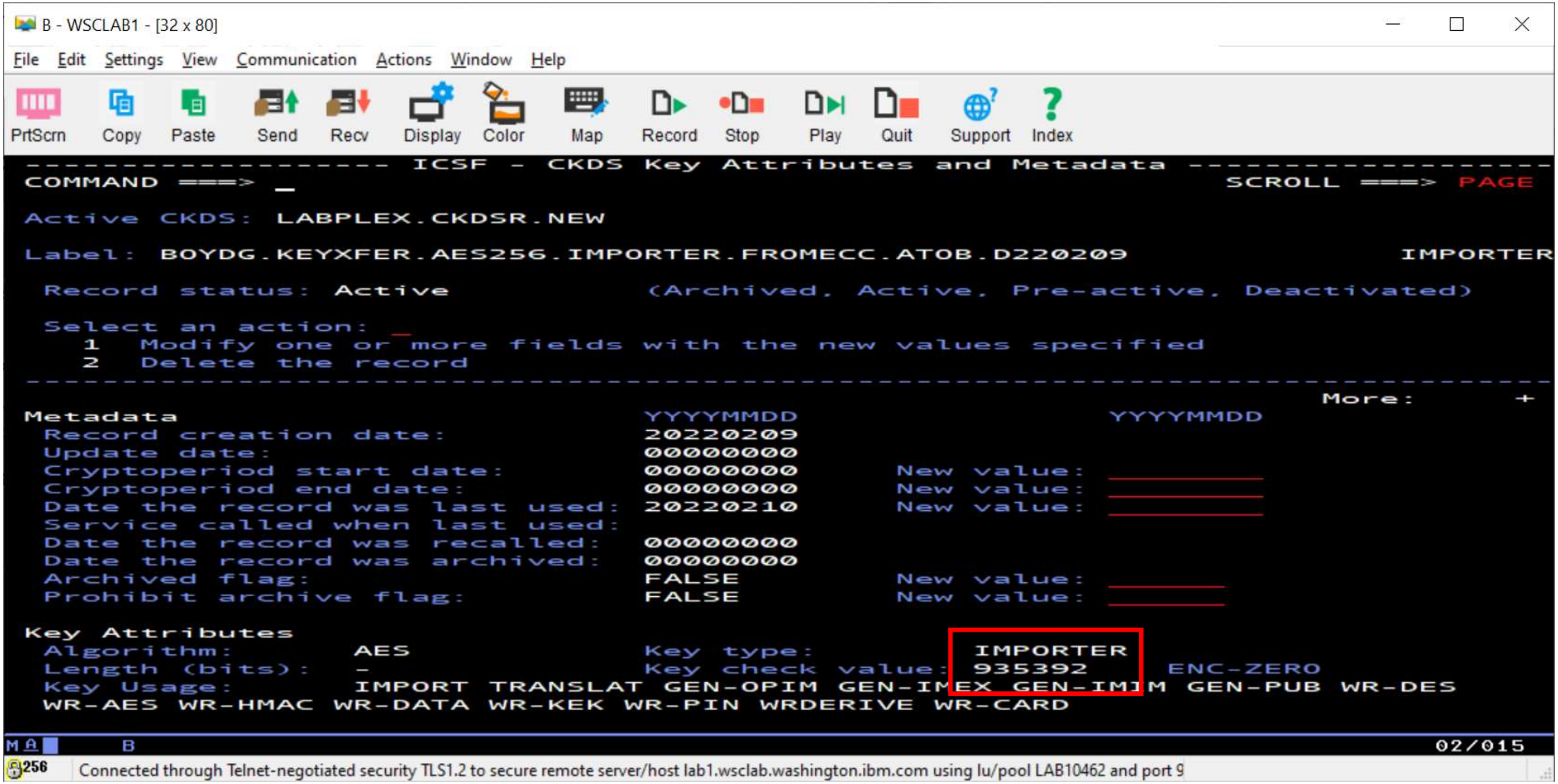
On A – Exporter key KCV=935392



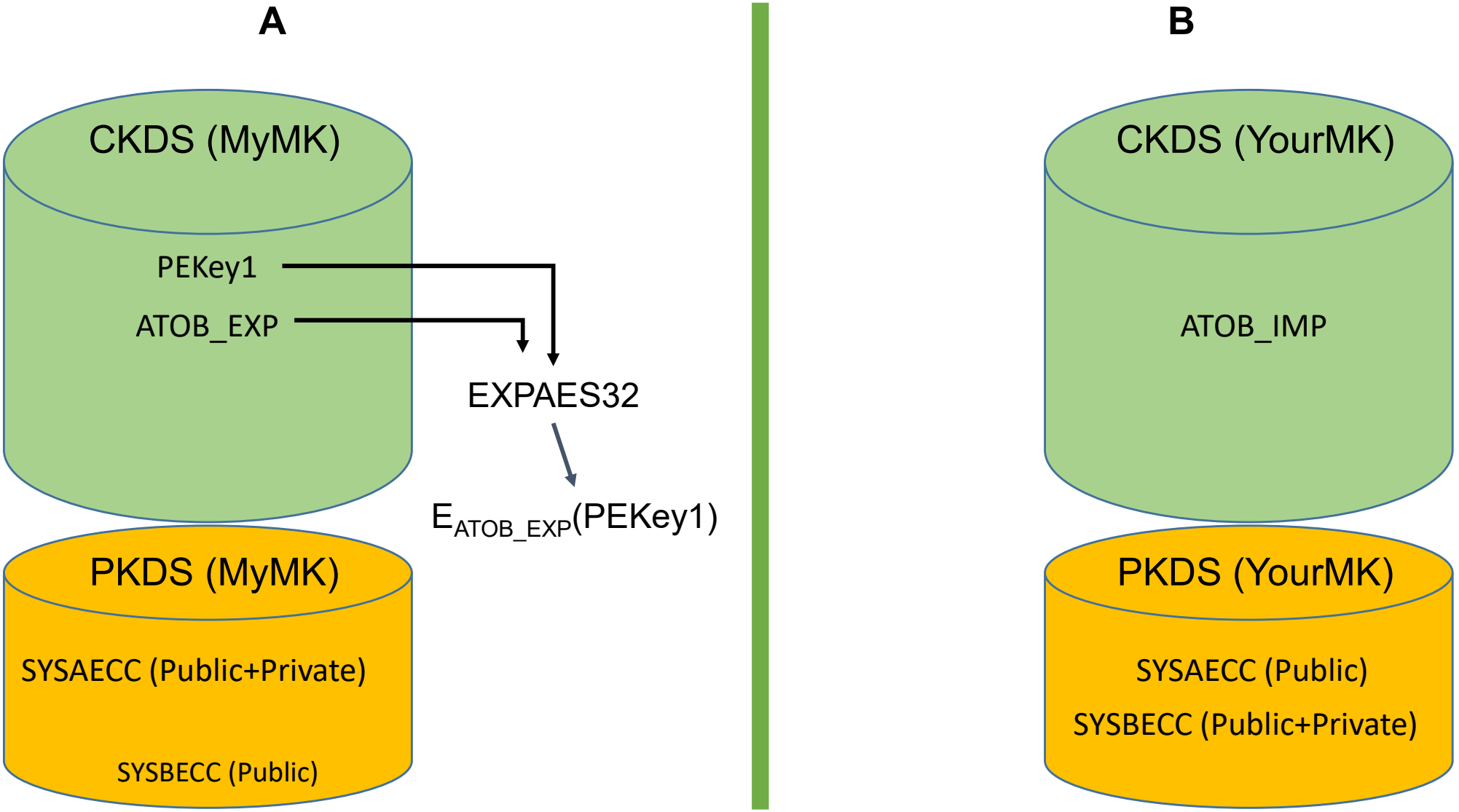
Create AES Importer on B



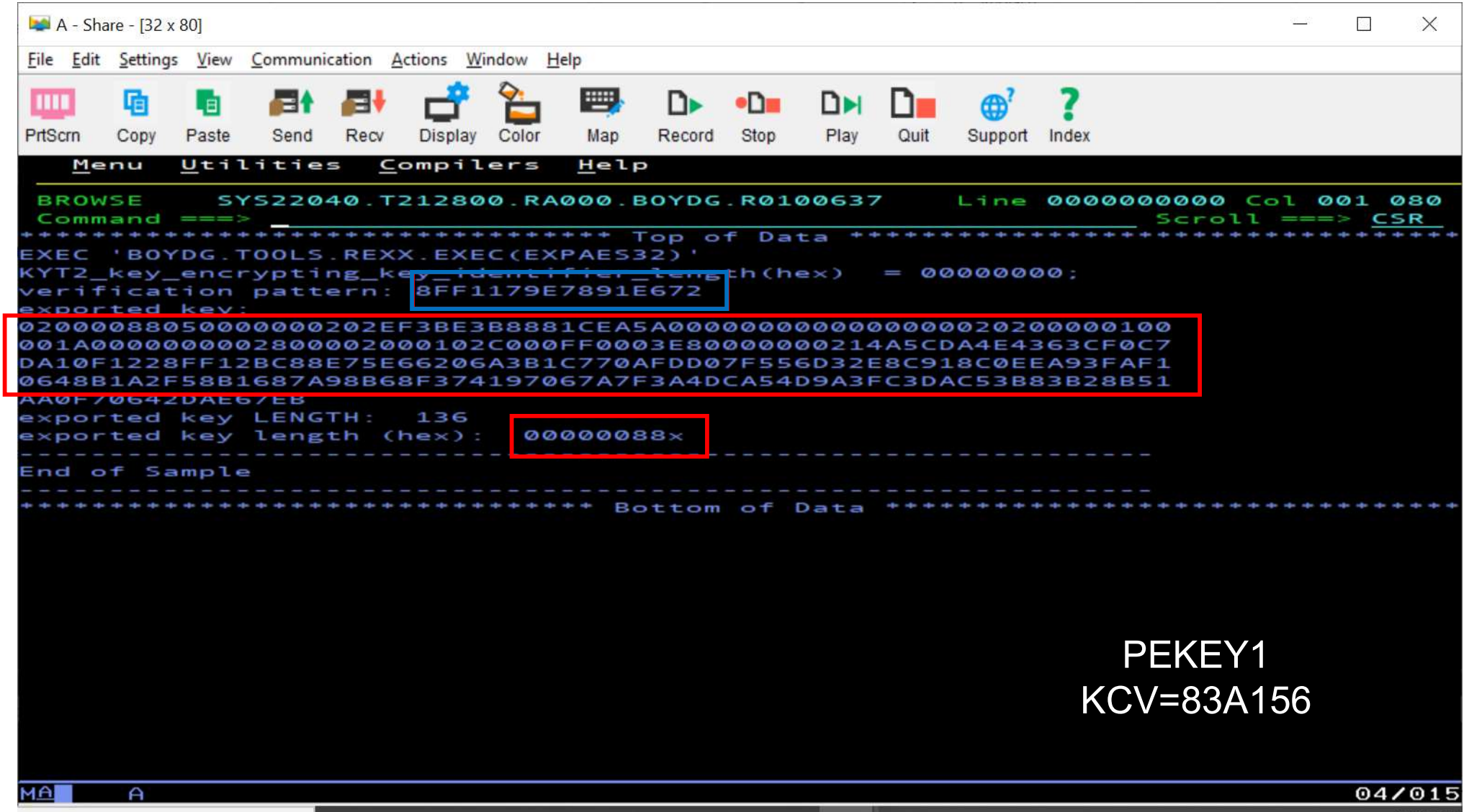
On B – Importer key KCV=935392



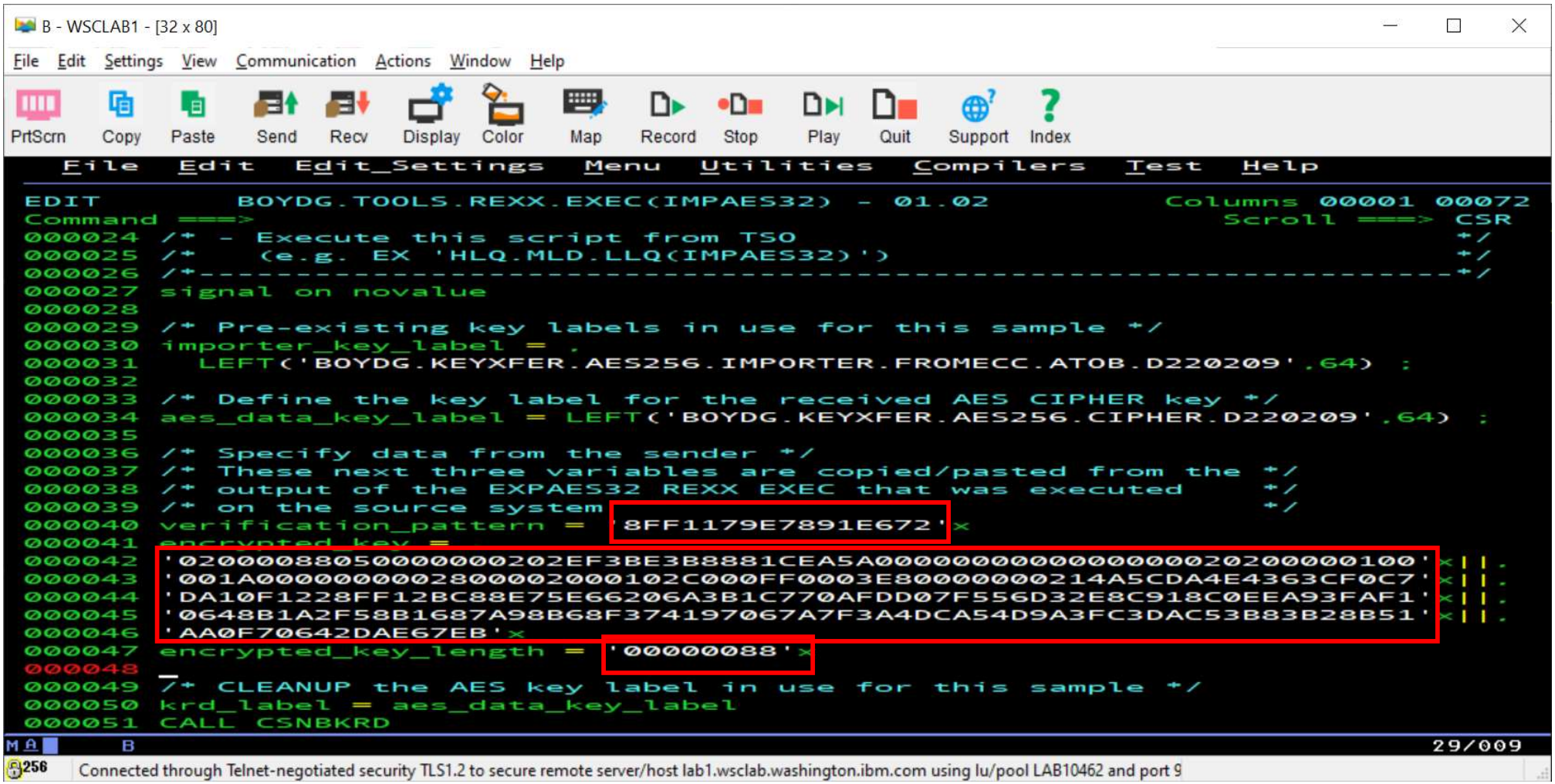
A: Export PEKey1 & B: Import PEKey1



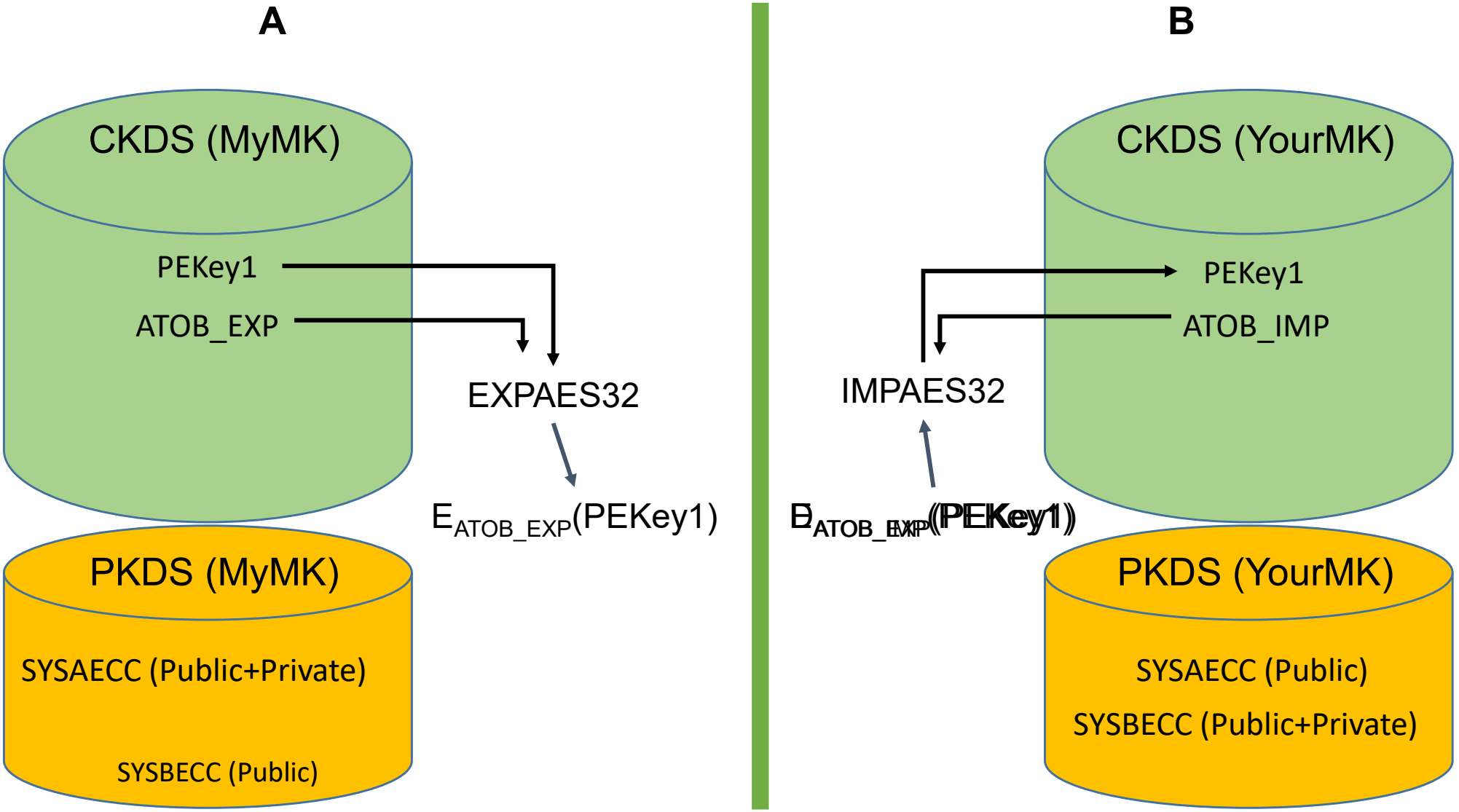
On A – Export the CIPHER key



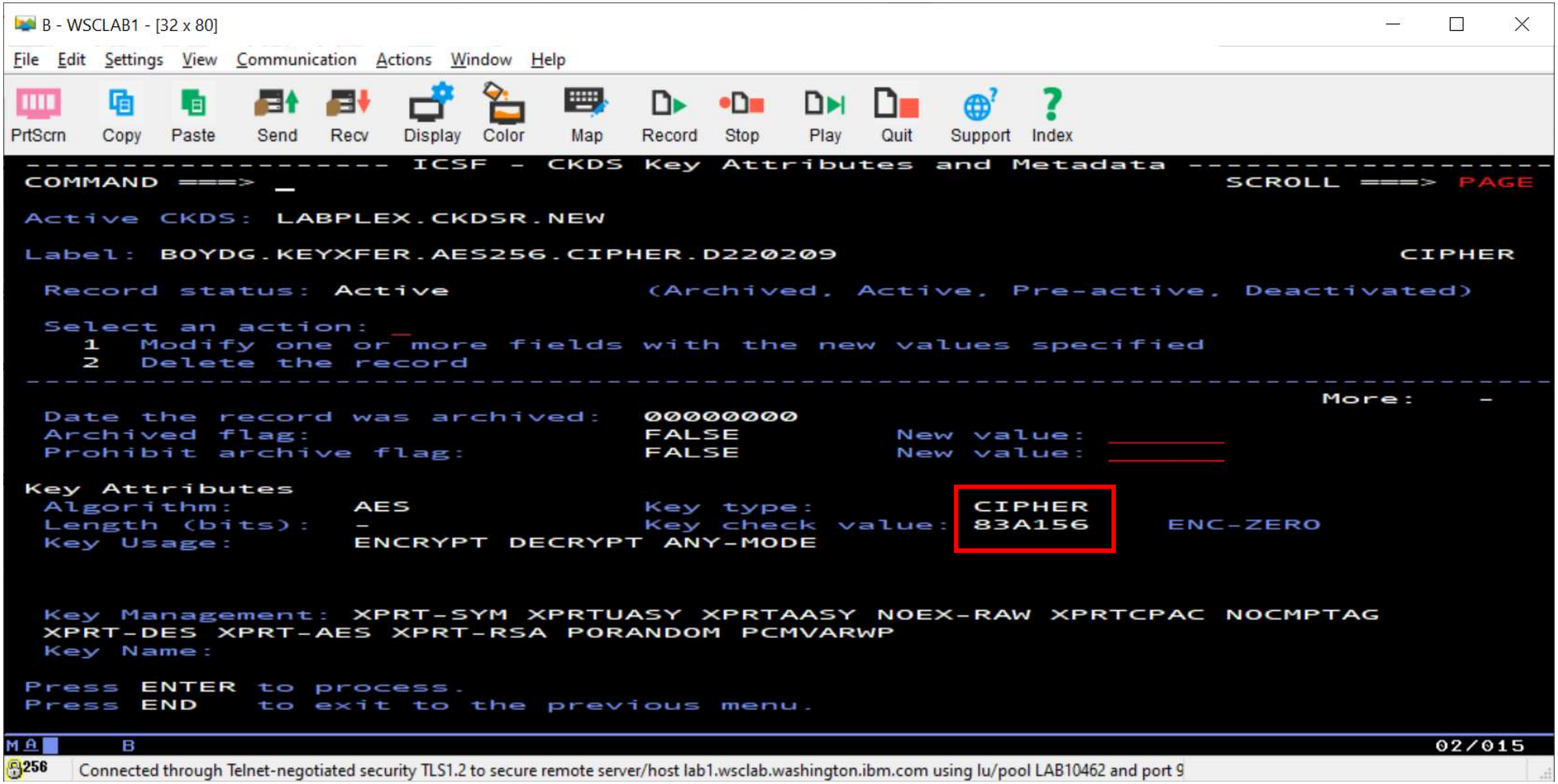
On B – Import the CIPHER key



A: Export PEKey1 & B: Import PEKey1



On B – Our copy of the CIPHER key



Summary

- With common master keys, process is simpler, but you may have to bypass some of the security protocols
- Establishing the EXPORTER/IMPORTER keys between two systems does not have to be done every time you need to transfer keys
- There are now tools (CKDS/PKDS Keys Utility) to validate the process

- My goal here was not to intimidate …
  - Key Management is important

References

- TechDoc - Transporting AES encrypted data keys from one z/OS host to another by Philippe Richard
  - https://www.ibm.com/support/pages/system/files/inline-files/Transporting_AES_DATA_keys_new.pdf
- IBM Community - Transporting AES Keys
  - https://community.ibm.com/community/user/ibmz-and-linuxone/viewdocument/drvaesmp?CommunityKey=6593e27b-caf6-4f6c-a8a8-10b62a02509c&tab=librarydocuments
- GitHub KeyXfer
  - http://ftpmirror.your.org/pub/misc/ftp.software.ibm.com/s390/zos/tools/keyxfer/

Questions