

# What keyring? What certificates? All I know is TLS works only in Server Authentication!

Wai Choi

IBM

November 4<sup>th</sup> 2021

Session 1AQ



## Agenda

- Review of what we learnt about how to set up server and client keyrings for TLS for Server authentication – What keyring? What certificates? All I know is TLS doesn't work

(<https://conferences.gse.org.uk/2020/agenda/presentation/1912>)

- Overview on client authentication process which includes
  - Handshake
  - ID mapping
- Steps to tackle a certificate related handshake and ID mapping problem in TLS in Client authentication



# Quick review on Server Authentication



## Server and client set up keyrings for Server authentication

- Certificate must be placed in a key ring before it can be used by an application to perform identification and validation
- The server admin
  - sets up a key ring with the **whole chain** (this example shows a chain of 2):
    - the server certificate
    - the issuer CA certificate (it is also the root in this case)
  - sends the **root CA certificate** to the client admin (not the server certificate !!!)
- The client admin
  - sets up a key ring with the server's root CA certificate:
    - The server's root CA certificate (this is also the issuer certificate in this case)
- Notice that in the case of a chain of N, the server keyring should contain N certificates, but the client keyring only needs the root CA certificate no matter how long the chain is.

### Server keyring

- FTP Server cert 
- CA cert that signed FTP server cert



Same CA cert

- CA cert that signed FTP server cert

### Client keyring



## Steps to tackle from server side

- Find out which party is the **server**, which party is the **client**
- **Server side:**
  1. What is the **configuration file** which includes the keyring information?
  2. What is the **keyring name**? Who is the **keyring owner**?
  3. Does the keyring contain all the needed **certificates**?
  4. Which one is the server certificate? Who owns it?
    - Usage is **Personal**
    - Marked as **DEFAULT** (the most popular set up used by TLS)
  5. Does the server certificate have a **private key** associated with it and is its status **TRUST**?
  6. What ID will be using the keyring? Does it have **access** to the private key?
    - Access to keyring means access to certificates in the keyring, but not the access to their private keys
    - Simpler set up if the accessing ID is the owner of the certificate, and owner of the keyring
    - If the access control is through RDATA LIB, make sure it is active and raclisted



## Steps to tackle from client side

- **Client side:**

1. What is the **configuration file** which includes the keyring information?
2. What is the **keyring name**? Who is the **keyring owner**?
3. Are the certificates **CERTAUTH certificates**?
4. Which one is the **root CA** certificate of the server? Is its status **TRUST**?
5. What ID will be using the keyring? Does it have **access** to the keyring?
  - Access to keyring means access to certificates in the keyring
  - If the access control is through RDATA LIB, make sure it is active and raclisted



# Story continues...Client Authentication



## What is Client authentication

- Client authentication is an inseparable step following server authentication. You can't have client authentication by itself
- It is the server side to determine if client authentication is needed
- Base on the server authentication set up, here are the extra steps
- **The client admin**
  - connect the following certificates to the client's key ring with the whole chain (suppose this is a chain of 2):
    - the client certificate
    - the issuer CA certificate (it is also the root in this case)
  - send the client certificate and the client's issuer CA certificate(s) in a PKCS7 package to the server admin
    - The package helps the server admin to determine the right CA to use in the server keyring
- **The server admin**
  - connect the following certificate to the server's key ring:
    - the client's issuer CA certificate (it is also the root in this case)
  - add the client certificate (no private key) to RACF (no need to put it in keyring) under the ID to be mapped to the client

### Client RACF DB

#### Client keyring

- CA cert that signed FTP server cert \*
- FTP Client cert \*\*
- CA cert that signed FTP client cert \*



### Server RACF DB

#### Server keyring

- FTP Server cert \*\*
- CA cert that signed FTP server cert \*
- CA cert that signed FTP client cert \*



- FTP Client cert \*

\*\* Cert and private key

\* Cert only





## What happens in Client authentication

- Handshake process part 2
  - Server authentication ends after the client validated the server certificate with the server's root CA in the client keyring
  - Then the client sends the client certificate to the server for it to validate with the client's root CA in the server keyring
- Mapping process continues after the Handshake process
  - RACF uses the client's certificate to map to an ID that owns the certificate in the DB



# How to tackle Client authentication problems from a certificate perspective



## Handshake process part 2



## Steps to tackle from client side

- **Client side:**

1. Make sure Client keyring already set up correctly for server authentication (see last year's presentation)
2. Does the keyring contain all the needed certificates for client authentication?
3. Which one is the client certificate? Who owns it?
  - Usage is Personal
  - Marked as DEFAULT (the most popular set up used by TLS)
4. Does the client certificate have a private key associated with it?
5. Do the client certificate and all its issuers have TRUST status?
6. Does the ID used to access the keyring have access to the client's private key?
  - Access to keyring means access to certificates in the keyring, but not the access to their private keys (with client authentication, private key is involved)
  - The access level to the keyring done for server authentication step may need to be changed from READ to UPDATE, if the accessing ID is not the client cert owner
  - If the access control is through RDATA LIB, make sure it is refresh



## Example on tracing problem from client side

### Client side:

```
TTLSTLSKeyRingParms From PROFILE.TCPIP.CLIENT
{
  Keyring                XXClient/XXClientRing ①
}
```

### RACDCERT ID(XXClient) LISTRING(XXClientRing)

Digital ring information for user **XXClient**: ②

Ring:

>XXClientRing<

Certificate Label Name	Cert Owner	USAGE	DEFAULT
Server Root CA	CERTAUTH	CERTAUTH	NO
XXClient Cert	ID(XXClient)	PERSONAL	YES
XXClient Intermediate CA	CERTAUTH	CERTAUTH	NO
XXClient Root CA	CERTAUTH	CERTAUTH	NO

②


③



## RACDCERT ID(XXClient) LISTCHAIN(LABEL('XXClient Cert'))

Certificate 1:

Digital certificate information for user **XXClient**:

Label: XXClient Cert  
 Certificate ID: 2Qbmxcli2eXi4tNAw4WZo0BA  
 Status: **TRUST**   
 Start Date: 2021/10/01 01:00:00  
 End Date: 2022/09/30 00:59:59

5


...  
 Private Key: **YES**   
 Certificate Fingerprint (SHA256):

4

...  
 Ring Associations:  
 Ring Owner: XXClient  
 Ring:  
 >XXClientRing<

Certificate 2:

Digital certificate information for **CERTAUTH**:

Label: XXClient Intermediate CA  
 Certificate ID: 2QinxcLi2eYj4tMAw4WZo0BD  
 Status: **TRUST**   
 Start Date: 2015/02/17 01:00:00  
 End Date: 2025/12/31 00:59:59

5


...  
 Private Key: NO

Certificate Fingerprint (SHA256):

...  
 Ring Associations:  
 Ring Owner: XXClient  
 Ring:  
 >XXSClientRing<

Certificate 3:

Digital certificate information for **CERTAUTH**:

Label: XXClient Root CA  
 Certificate ID: 2QkkxcLi2eZj4tMAw4WZo0BE  
 Status: **TRUST**   
 Start Date: 2015/01/01 01:00:00  
 End Date: 2035/12/31 00:59:59

5

...  
 Private Key: NO  
 Certificate Fingerprint (SHA256):

...  
 Ring Associations:  
 Ring Owner: XXClient  
 Ring:  
 >XXClientRing<

Chain information:

Chain contains 3 certificate(s), **chain is complete**  
 Chain contains ring in common: **XXClient/XXClientRing**

2



**RLIST RDATALIB** `XXClient.XXClientRing.LST`

6

CLASS            NAME

-----

-----

RDATALIB        XXCLIENT.XXCLIENTRING.LST

LEVEL    OWNER            UNIVERSAL ACCESS    YOUR ACCESS    WARNING

-----

-----

-----

-----

-----

...

USER            ACCESS

-----

-----

XXCLIENT        READ

YYCLIENT        UPDATE ← if YYCLIENT accesses XXCLIENT's certificate,  
UPDATE is needed

...

**\*\* Make sure the RDATALIB class is refresh!!!**



SETR RACLIST(RDATALIB) REFRESH



## Steps to tackle from server side

- **Server side:**

1. Server keyring already set up for server authentication
2. Make sure client certificate and its issuers certificates are correctly received
  - Ideally, the client would send a PKCS7 package which contains the cert chain, from client cert to root
  - Make use of RACDCERT CHECKCERT on the package to ensure the right certs were received
  - If the client certificate and the issuers' CA certificates were sent individually, the server admin needs additional information from the client admin, like the fingerprint, the serial number, SDN, IDN... etc to make sure the right certs were sent
3. Which one is the root CA certificate of the client? Is its status TRUST?
4. **Is the client certificate installed in RACF under a regular RACF user ID? Is its status TRUST? – assuming the mapping mechanism is based on the cert installed**





RACDCERT CHECKCERT(<dataset contains the PKCS7 package>)

## Example on tracing problem from server side

Certificate 1:

Start Date: 2021/10/01 01:00:00

End Date: 2022/09/30 00:59:59

...

Issuer's Name:

<Intermediate SDN>

Subject's Name:

<Client SDN>

...

Certificate Fingerprint (SHA256):

...

Certificate 2:

Start Date: 2015/02/17 01:00:00

End Date: 2025/12/31 00:59:59

...

Issuer's Name:

<Root SDN>

Subject's Name:

<Intermediate SDN>

...

Certificate Fingerprint (SHA256):

...

If the cert in the package already added in RACF DB, you will see the owner, the label, the certificate ID and the status, eg Certificate 3 (root) entry would show this

Certificate 3:

Start Date: 2015/01/01 01:00:00

End Date: 2035/12/31 00:59:59

...

Issuer's Name:

<Root SDN>

Subject's Name:

<Root SDN>

...

Certificate Fingerprint (SHA256):

...

Chain information:

Chain contains 3 certificate(s), **chain is complete**

2

Certificate 3:

Digital certificate information for **CERTAUTH**:

**Label:** Root CA from XXClient

**Certificate ID:** 1PkkxcLi2eZj4tMAw4WZo0BE

**Status:** TRUST

Start Date: 2015/01/01 01:00:00

End Date: 2035/12/31 00:59:59

...



## Server side:

```
TTLSTLSKeyRingParms
{
  Keyring                               XXServer/XXServerRing
}
RACDCERT ID(XXServer) LISTRING(XXServerRing) ①
```

From PROFILE.TCPIP.SERVER

Digital ring information for user **XXServer**:

Ring:

```
>XXServerRing<
Certificate Label Name          Cert Owner          USAGE          DEFAULT
-----
SSL Cert                       ID (XXServer)     PERSONAL       YES
Local Intermediate CA         CERTAUTH           CERTAUTH       NO
Local Root CA                 CERTAUTH           CERTAUTH       NO
Client Root CA                CERTAUTH           CERTAUTH       NO
```


③

Notice that no client cert is in the keyring



### RACDCERT CERTAUTH LIST(LABEL('Client Root CA'))

Digital certificate information for  
**CERTAUTH:**


Label: Client Root CA  
 Certificate ID: 2QkkxcLi2eZj4tMAw4WZo0BE  
 Status: **TRUST**   
 Start Date: 2015/01/01 01:00:00  
 End Date: 2035/12/31 00:59:59  
 ...  
 Private Key: NO  
 ...  
 Certificate Fingerprint (SHA256):  
 ...  
 Ring Associations:  
 Ring Owner: XXServer  
 Ring:  
 >XXServerRing<  
 Chain information:  
 Chain contains 1 certificate(s), chain  
 is complete  
 Chain contains ring in common:  
 XXServer/XXServerRing

3

Note: The label 'Client Root CA' doesn't need to be the same as used in the Client system

### RACDCERT ID(Bob) LIST(LABEL("Bob's cert"))

Digital certificate information for user  
**Bob:**

Label: Bob's cert  
 Certificate ID: 2QmmxcLi2eZj4tMAw4WZo0BE  
 Status: **TRUST**   
 Start Date: 2021/10/01 01:00:00  
 End Date: 2022/09/30 00:59:59  
 ...  
 Private Key: NO  
 ...  
 Certificate Fingerprint (SHA256):  
 ...

4

#### Note:

- 1) The label "Bob's cert" and the cert owner ID, Bob, do not need to be the same as used in the Client system. It is XXClient in the client's system
- 2) No private key is needed
- 3) It is not connected to a keyring



## ID Mapping process



## ID Mapping process after Handshake

- Handshake process part 2
  - Server authentication ends after the client validated the server certificate with the server's root CA in the client keyring
  - Then the client sends the client certificate to the server for it to validate with the client's root CA in the server keyring
  - Mapping process continues after the Handshake process
- ID Mapping process continues after the Handshake process
  - In fact there are three mapping ways, and what we discussed is just the first way
    1. RACF finds an ID that **owns** the same certificate in RACF's DB as the client's certificate
    2. RACF finds an ID that owns a certificate mapping profile in RACF's DB, which matches the subject and/or issuer distinguished name in the client's certificate. This is known as Certificate Name Filtering
    3. RACF extracts the ID from the client's certificate's HostIDMapping extension
  - The order of operation is as listed above, ie, only if no matching cert is found, the certificate mapping will be used; HostIdMapping will be used only if no matching certificate nor matching certificate mapping profile is found



## Steps to tackle from server side if Certificate Mapping is used

- **Server side:**

1. Make sure the server side has received the following information on the client's certificate:

- Subject Distinguished Name (SDN) or
- Issuer Distinguished Name (IDN) or
- both

2. Is there a certificate mapping filter in RACF?

3. Is the filter TRUST?

4. Is the DIGTNMAP class raclisted and refreshed?



## Example to track issues if Certificate Mapping is used

### • Server side:

1. Find out, from the client, the Subject Distinguished Name (SDN) or Issuer Distinguished Name (IDN) or both on the client's certificate, e.g.

- SDN: CN=Bob.OU=DeptC.C=UK, IDN: OU=TestGroup.C=US

2. Find if there is certificate mapping profile (aka certificate name filter) containing the matching information above

- DBUNLOAD record type 508 contains information on certificate name filter
- Use DFSORT ICETOOL to create a report. You may find entries like these

User ID	Label	Filter
USER1	USRER1MAP	CN=User1 OU=DeptA C=US
GROUP1	GROUP1MAP	¢CN=OU=DeptB C=US
<b>BOB</b>	<b>BOBMAP</b>	CN=OU=TestGroup.C=US¢CN=Bob.OU=DeptC.C=UK

3. Issue RACDCERT ID(**BOB**) LISTMAP(LABEL('BOBMAP')) to check if its status is TRUST

Mapping information for BOB:

Label: BOBMAP

Status: TRUST 

Issuer's Name Filter:

> OU=TestGroup.C=US <

Subject's Name Filter:

> CN=Bob.OU=DeptC.C=UK <

**\*\* Make sure the DIGTNMAP class is refresh!!!**

SETR RACLIST(DIGTNMAP) REFRESH



## Steps to tackle from server side if HostIdMapping is used

- **Server side:**

- 1. Make sure the server side has received the following information:**

- the correct user ID and host name as shown in the client's cert's HostIdMapping extension
- the issuer certificate

*Note: In the case of a certificate chain of 3, the issuer certificate of the client is the intermediate CA, not the root CA*

- 2. Is the profile in SERVAUTH class set up correctly?**

- 3. Does the ID have READ access to the SERVAUTH profile?**

- 4. Is the issuer certificate installed in RACF and it has HIGHTRUST status?**





## Example to track issues if HostIDMapping is used

- **Server side:**

- Need to find out, from the client, the user ID and host name in the HostIDMapping extension, e.g.

1

**Bob@Host1**

- Find if Bob has READ access to profile IRR.HOST.<host name> in SERVAUTH class

RLIST SERVAUTH IRR.HOST.HOST1

CLASS NAME

-----

SERVAUTH IRR.HOST.HOST1

LEVEL OWNER UNIVERSAL ACCESS YOUR ACCESS WARNING

-----

...

USER ACCESS

-----

**BOB** READ



3

2

4

- Issue RACDCERT CERTAUTH LIST(LABEL('Client Intermediate CA')) to check if the client's issuer certificate status is HIGHTRUST

Digital certificate information for CERTAUTH:

Label: Client Intermediate CA

Certificate ID: 2QinxLi2eYj4tMAw4WZo0BD

Status: **HIGHTRUST**

Start Date: 2015/02/17 01:00:00

End Date: 2025/12/31 00:59:59

...

Private Key: NO

Note: This CA cert does not need to connect to any keyring



## Pros and Cons on the 3 ways of ID mapping in Client Authentication

Mapping using	Pros	Cons
Client certificate	<ul style="list-style-type: none"> <li>• Simpler to set up</li> </ul>	<ul style="list-style-type: none"> <li>• Need to install client cert in the server system</li> <li>• Administrative cost could be high to install a large number of client certificates in the server side</li> </ul>
Certificate Name Filter	<ul style="list-style-type: none"> <li>• Can map multiple certs to a single ID</li> <li>• A preferred option when a large number of users need to map to a group ID</li> </ul>	<ul style="list-style-type: none"> <li>• Need to communicate SDN/IDN information to server system</li> <li>• Server needs to set up the mapping profiles</li> </ul>
HostIdMapping	<ul style="list-style-type: none"> <li>• Can map one cert to different IDs in different systems</li> </ul>	<ul style="list-style-type: none"> <li>• Need to communicate HostIdMapping extension info to the server side</li> <li>• Server needs to install issuer CA and SERVAUTH profiles</li> </ul>



## Useful information related to Certificate Name Filtering and HostIdMapping



## Sample job using ICETOOL to find out the filters from DBUNLOAD flat file output

```
//MYJOBICE JOB CLASS=5,MSGCLASS=H,NOTIFY=&SYSUID,MSGLEVEL=0
//*-----
//RACFICE EXEC PGM=ICETOOL,PARM='MSGPRT=ALL'
//TOOLMSG DD SYSOUT=*
//PRINT DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//DBUDATA DD DISP=SHR,DSN=MYTEST.FORMAP.IRRDBU00.FLATFILE
//TEMP001 DD DISP=(NEW,DELETE,DELETE),SPACE=(CYL,(1,5,0)),
// UNIT=SYSALLDA
//TOOLIN DD *
SORT FROM(DBUDATA) TO(TEMP001) USING(CERT)
DISPLAY FROM(TEMP001) LIST(PRINT) -
PAGE -
TITLE('Certificate Mappings') -
DATE(4MD/) -
TIME(12:) -
BLANK -
ON(308,08,CH) HEADER('User ID') -
ON(266,32,CH) HEADER('Label') -
ON(555,255,CH) HEADER('Filter')
SORT FROM(DBUDATA) TO(TEMP001) USING(CMDS)

//CERTCNTL DD *
SORT FIELDS=COPY
INCLUDE COND=(5,4,CH,EQ,C'0508')
OPTION VLSHRT
//CMDSCNTL DD *
SORT FIELDS=COPY
INCLUDE COND=(5,4,CH,EQ,C'0508')
OUTREC FIELDS=(1,4,C'RACMAP LIST ID(',308,8,C') is ',
30,8,C' and is assigned to ',79,20)
OPTION VLSHRT
```

User ID

Label

Filter

ç is a separator between IDN and SDN filters

**BOB**

**BOBMAP**

OU=TestGroup.C=USçCN=Bob.OU=DeptC.C=UK

**GROUPA**

**GRPAMAP**

OU=Test DeptA.C=DEç

Trailing ç indicates IDN filter only



# Using PKI Services to request cert with HostIdMapping extension

## 1-Year PKI Generated Key Certificate

Choose one of the following:

- Request a New Certificate

Enter values for the following field(s)

Enter the requestor's email address

Pass phrase for securing this request. You will need to supply this value v

Reenter your pass phrase to confirm

Common Name

HostIdMapping Extension value in subject-id@host-name form

HostIdMapping Extension value in subject-id@host-name form

Email address for distinguished name MAIL= attribute (optional)

Select the key type and key size

Request a cert from PKI with HostIdMapping extension

Use this cert for client authentication in systems Host1 and Host2

## Single Issued Certificate

<b>Requestor:</b>	robert_lee@us.ibm.com	<b>Created:</b>	2021/09/29
<b>Status:</b>	Active	<b>Modified:</b>	2021/09/29
<b>Template:</b>	1-Year PKI Generated Key Certificate	<b>PassPhrase:</b>	[REDACTED]
<b>Serial #:</b>	4		
<b>Previous Action Comment:</b>	Issued certificate		
<b>Archived KeyId:</b>	E7AAB448C4DEDA2CE5AEECE61E01FF44F96BB91C		
<b>Subject:</b>	CN=HostIdCert for Bob,OU=Class 1 Internet Certificate CA,O=The Firm		
<b>Issuer:</b>	OU=Subordinate CA 1,O=IBM,C=US		
<b>Validity:</b>	2021/09/29 00:00:00 - 2022/09/28 23:59:59		
<b>Usage:</b>	handshake(digitalSignature, keyEncipherment)		
<b>Extended Usage:</b>	not specified		
<b>HostIdMap:</b>	Bob@Host1 Robert@Host2		
<b>Key type and key size:</b>	RSA-2048		
<b>Signature algorithm:</b>	sha-256WithRSAEncryption		
<b>SHA256 fingerprint:</b>	A4:53:5E:83:43:E9:20:83:24:55:4F:2A:C9:D3:C8:DE:C8:B3:9E:70:5C:CE:2F:6A:93:99:1C:C1:5C:6B:FF:A1		



How much do you remember?



1. Client authentication can be set up without server authentication. True or False?

False

2. Which certificate is needed in the server's keyring specific for client authentication handshake process? Client cert or Client root CA cert

Client root CA cert

3. Does the server side need to add the client certificate to the RACF DB? Yes or No

It depends. Yes, if matching is based on an existing client certificate

4. What are the two other ways to find an ID corresponds to the client certificate which is being used during handshake?

Certificate Name Filtering

HostIdMapping

5. Can you create a certificate with HostIdMapping extension with RACDCERT?

No. You need to use z/OS PKI Services

6. All the steps are checked OK, but I still have the problem. What may be a possible cause?

The application has not been refreshed to pick up the keyring changes



## References

- **Cryptographic Server Manual**

**Cryptographic Services PKI Services Guide and Reference**

[https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4sa232286/\\$file/ikya100\\_v2r4.pdf](https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4sa232286/$file/ikya100_v2r4.pdf)

**Cryptographic Services System Secure Sockets Layer Programming**

[https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4sc147495/\\$file/gska100\\_v2r4.pdf](https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4sc147495/$file/gska100_v2r4.pdf)

- **Security Server Manuals:**

**RACF Command Language Reference**

[https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4sa232292/\\$file/icha400\\_v2r4.pdf](https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4sa232292/$file/icha400_v2r4.pdf)

**RACF Security Administrator's Guide**

[https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4sa232289/\\$file/icha700\\_v2r4.pdf](https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R4sa232289/$file/icha700_v2r4.pdf)

- **RFCs**

**RFC5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile**

<https://tools.ietf.org/html/rfc5280>





## References

- **IBM Enterprise Knights videos on digital certificates:**

<https://ek-ibmz.mybluemix.net/video/c57660745a547e504d54793083a97b0d>

<https://ek-ibmz.mybluemix.net/video/d399cee97db684bbf4f0f4e2b42cff15>

- IBM Hot Topics

Issue #29: Drowning in digital certificates? Here's a lifeline!

<http://publibfp.dhe.ibm.com/epubs/pdf/e0z3n110.pdf>

Issue #21: RACDCERT tipbits. x509 digital certificate technology

<http://publibz.boulder.ibm.com/epubs/pdf/e0z2n1a0.pdf>

Issue #19: Grow your own. Using locally generated digital certificates

<http://publibz.boulder.ibm.com/epubs/pdf/e0z2n190.pdf>

Issue #14: Security alert: Do you want to proceed?

<http://publibz.boulder.ibm.com/epubs/pdf/e0z2n161.pdf>



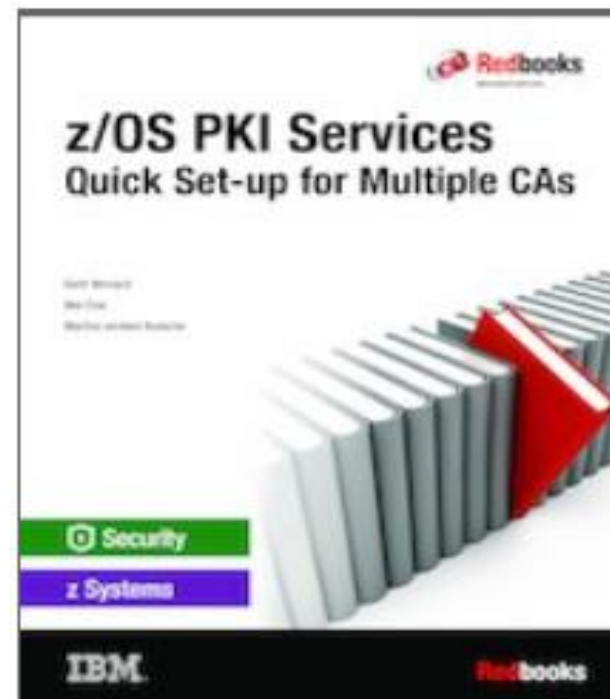
- **IBM PKI Redbooks**

- Managing Digital Certificates across the Enterprise**

- <https://www.redbooks.ibm.com/abstracts/sg248336.html?Open>

- z/OS PKI Services: Quick Set-up for Multiple CAs**

- <https://www.redbooks.ibm.com/abstracts/sg248337.html?Open>



Any questions?


# Please submit your session feedback!

- Do it online at <https://conferences.gse.org.uk/2021/feedback/1AQ>


- This session is 1AQ



1. What is your conference registration number?


 This is the three digit number on the bottom of your delegate badge

2. Was the length of this presentation correct?

 1 to 4 = "Too Short" 5 = "OK" 6-9 = "Too Long"


1 2 3 4 5 6 7 8 9

3. Did this presentation meet your requirements?

 1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

4. Was the session content what you expected?

 1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9



# Become a member of GSE UK

- Company or individual membership available
- Benefits include:
  - GSE Annual Conference: Receive 5 free places + 2 free places for trainees
  - 20% discount on fees for IBM Technical Conferences
  - 20% on IBM Training Courses in Europe
  - 15% discount for IBM STG Technical Conferences in the USA
  - 20% discount on the fee for taking the Mainframe Technology Professional (MTP) exams
  - European events – via GSE HQ
- Contact [membership@gse.org.uk](mailto:membership@gse.org.uk) for details



# GSE UK Conference 2021 Charity Raffle

- The GSE UK Region team hope that you find this presentation and others that follow useful and help to expand your knowledge of z Systems.
- Please consider showing your appreciation by kindly donating to our charities this year, Royal National Lifeboat Institution (RNLI) & Guide Dogs for the Blind. Then follow the link on your receipt to enter your receipt number & amount donated into the GSE Raffle. You will get a raffle entry for every pound donated.
- Follow the link below or scan the QR Code:

<http://uk.virginmoneygiving.com/GuideShareEuropeUKRegion>



Supporting

