# Digital Certificates Tidbits

## New York RACF User Group
## May 11th 2010

Wai Choi, CISSP
IBM Corporation
RACF/PKI Development & Design
Poughkeepsie, NY

e-mail: wchoi@us.ibm.com

# Trademarks

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

- CICS*
- DB2*
- IBM*
- IBM (logo)*
- OS/390*
- RACF*
- Websphere*
- z/OS*

* Registered trademarks of IBM Corporation

**The following are trademarks or registered trademarks of other companies.**

Identrus is a trademark of Identrus, Inc
VeriSign is a trademark of VeriSign, Inc
Microsoft, Windows and Windows NT are registered trademarks of Microsoft 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.

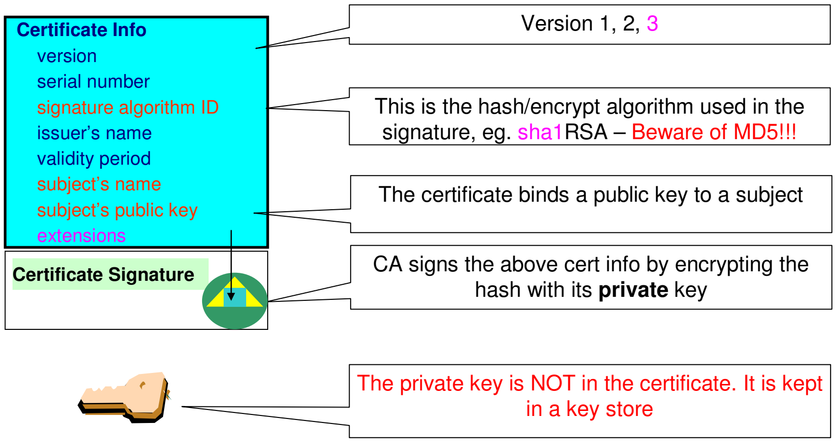
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products 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.

# Agenda

- Brief introduction to digital certificates
- Digital certificates support provided by z/OS
- The key ring set up
- Certificate management considerations
- PKI Services

# What's inside a Certificate?



**Certificate Info**
- version
- serial number
- signature algorithm ID
- issuer's name
- validity period
- subject's name
- subject's public key
- extensions

**Certificate Signature**

Version 1, 2, 3

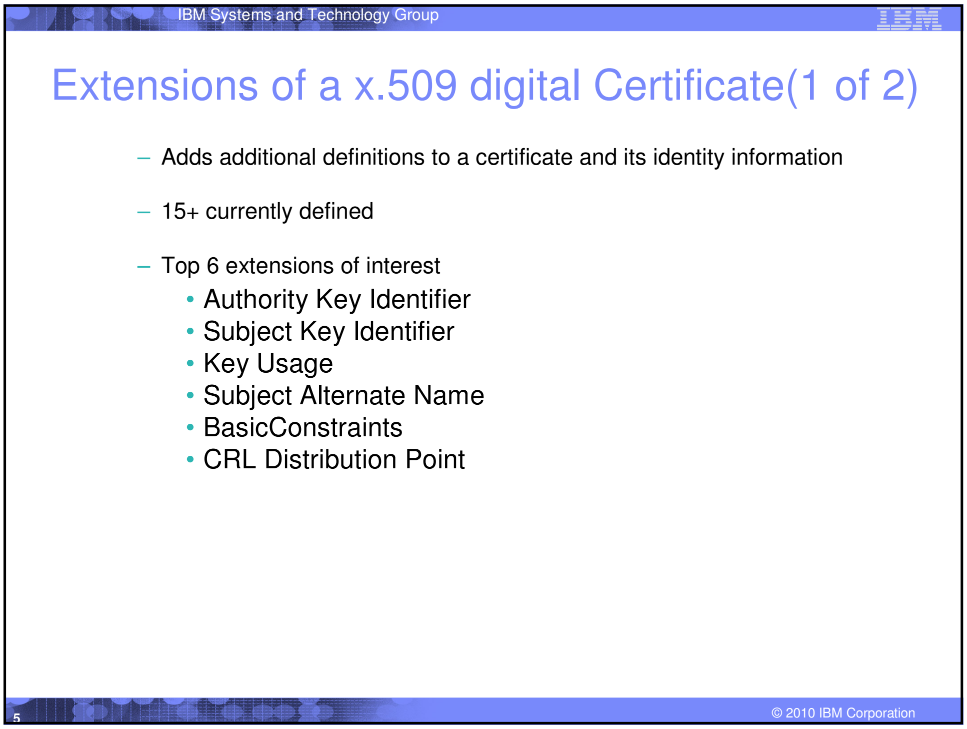
This is the hash/encrypt algorithm used in the signature, eg. sha1RSA – Beware of MD5!!!

The certificate binds a public key to a subject

CA signs the above cert info by encrypting the hash with its **private** key

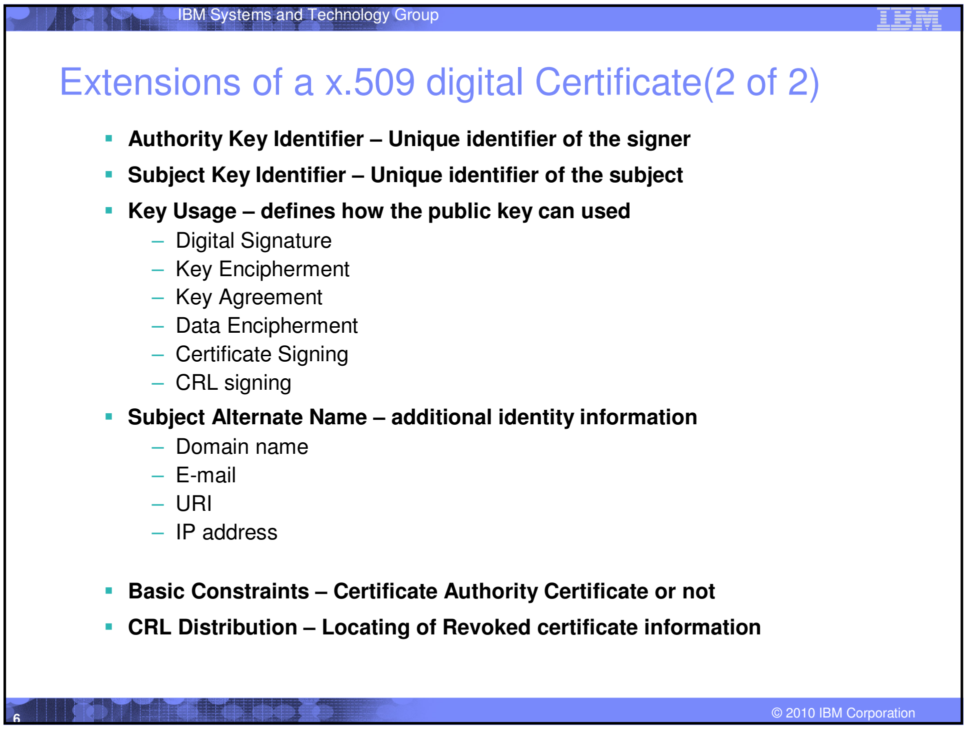
The private key is NOT in the certificate. It is kept in a key store

You can NOT change ANY of the certificate information!



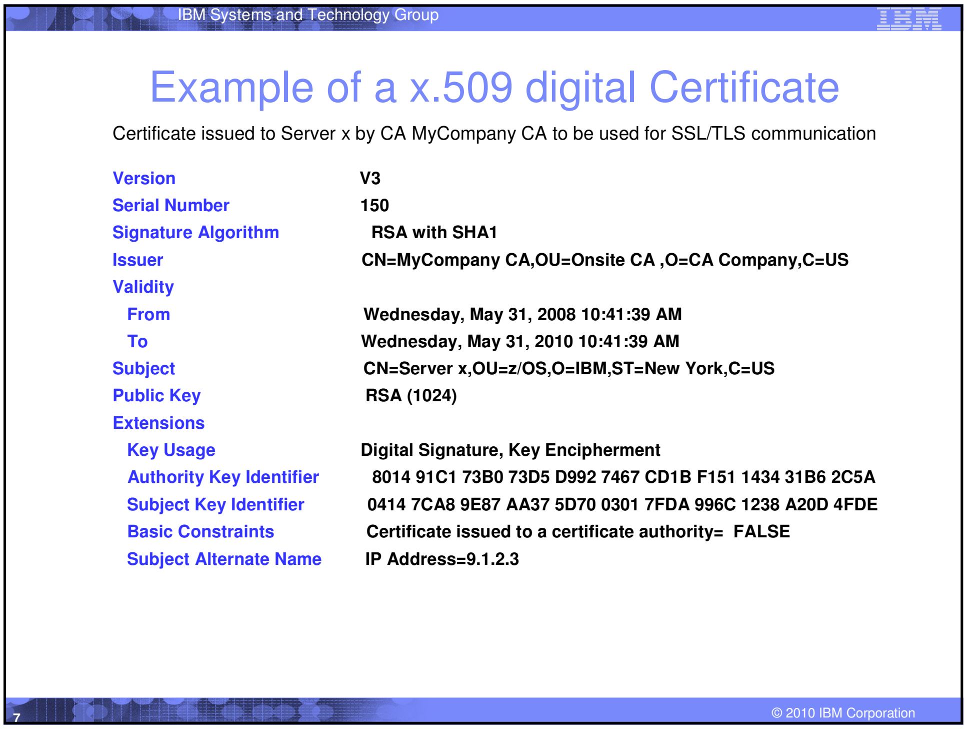
# Extensions of a x.509 digital Certificate(1 of 2)

- Adds additional definitions to a certificate and its identity information
- 15+ currently defined
- Top 6 extensions of interest
  - Authority Key Identifier
  - Subject Key Identifier
  - Key Usage
  - Subject Alternate Name
  - BasicConstraints
  - CRL Distribution Point



# Extensions of a x.509 digital Certificate(2 of 2)

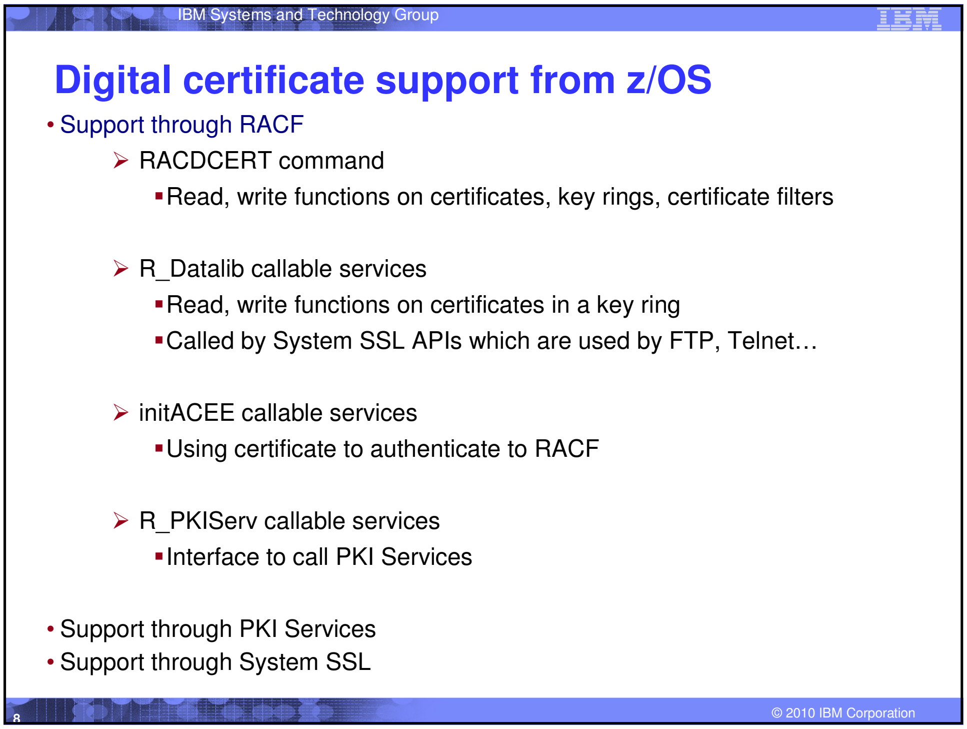
- **Authority Key Identifier – Unique identifier of the signer**
- **Subject Key Identifier – Unique identifier of the subject**
- **Key Usage – defines how the public key can used**
  - Digital Signature
  - Key Encipherment
  - Key Agreement
  - Data Encipherment
  - Certificate Signing
  - CRL signing
- **Subject Alternate Name – additional identity information**
  - Domain name
  - E-mail
  - URI
  - IP address
- **Basic Constraints – Certificate Authority Certificate or not**
- **CRL Distribution – Locating of Revoked certificate information**



# Example of a x.509 digital Certificate

Certificate issued to Server x by CA MyCompany CA to be used for SSL/TLS communication

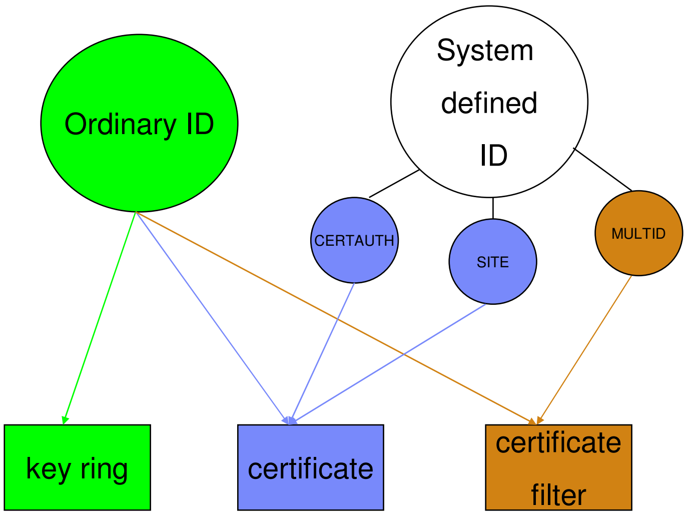
| | |
|---|---|
| **Version** | **V3** |
| **Serial Number** | **150** |
| **Signature Algorithm** | **RSA with SHA1** |
| **Issuer** | **CN=MyCompany CA,OU=Onsite CA ,O=CA Company,C=US** |
| **Validity** | |
| **From** | **Wednesday, May 31, 2008 10:41:39 AM** |
| **To** | **Wednesday, May 31, 2010 10:41:39 AM** |
| **Subject** | **CN=Server x,OU=z/OS,O=IBM,ST=New York,C=US** |
| **Public Key** | **RSA (1024)** |
| **Extensions** | |
| **Key Usage** | **Digital Signature, Key Encipherment** |
| **Authority Key Identifier** | **8014 91C1 73B0 73D5 D992 7467 CD1B F151 1434 31B6 2C5A** |
| **Subject Key Identifier** | **0414 7CA8 9E87 AA37 5D70 0301 7FDA 996C 1238 A20D 4FDE** |
| **Basic Constraints** | **Certificate issued to a certificate authority= FALSE** |
| **Subject Alternate Name** | **IP Address=9.1.2.3** |



# Digital certificate support from z/OS

- Support through RACF
  - ➢ RACDCERT command
    - ▪Read, write functions on certificates, key rings, certificate filters
  - ➢ R_Datalib callable services
    - ▪Read, write functions on certificates in a key ring
    - ▪Called by System SSL APIs which are used by FTP, Telnet…
  - ➢ initACEE callable services
    - ▪Using certificate to authenticate to RACF
  - ➢ R_PKIServ callable services
    - ▪Interface to call PKI Services
- Support through PKI Services
- Support through System SSL

# Relationship between entities and owner IDs



Ordinary ID

System defined ID

CERTAUTH

SITE

MULTID

key ring

certificate

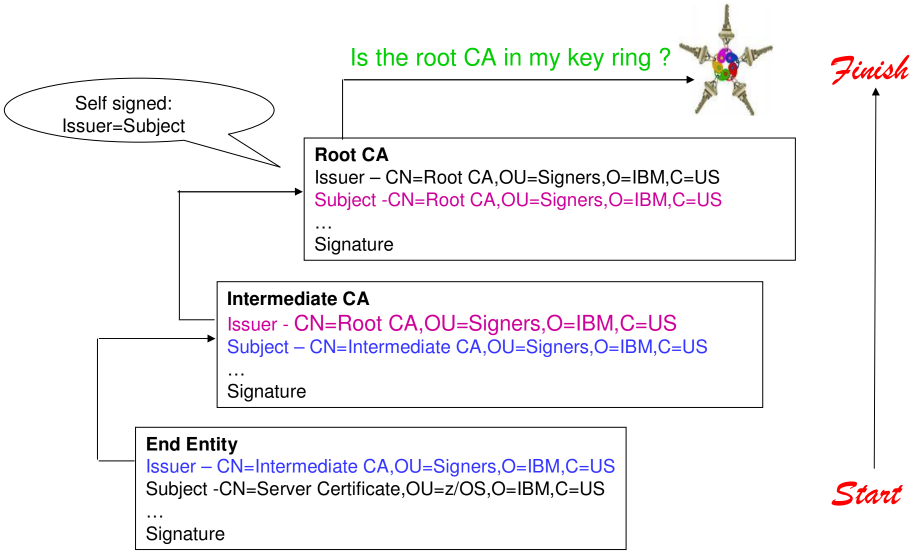
certificate filter

# Relationship between Certificate and Key ring

- ❑ Certificate must be placed in a key ring before it can be used by other middleware through R_datalib
- ❑ Three types of certificates in a ring that the middleware can utilize:
  - ➢ **Personal certificate (for identification)**
    - Under ordinary MVS ID or with USAGE PERSONAL
    - Its private key is also known to RACF
    - Sent to the client when SSL is initiated
  - ➢ **Certificate Authority certificate (for validation)**
    - Under CERTAUTH ID or with USAGE CERTAUTH
    - Its private key is not known to RACF
    - Used to authenticate the incoming certificate
    - eg cert A->cert B->cert C, in order to validate C, B and A need to be in the ring
  - ➢ **SITE certificate**
    - Under SITE ID or with USAGE SITE
    - For identification
      - Similar to personal certificate
      - But its private key can be shared (usual way to share key before V1R9)
    - For validation
      - Its private key is not known to RACF
      - Used to authenticate the incoming certificate without the complete chain
      - eg. Using the above example, A is not required to be in the ring if B is a SITE certificate.

!!!Note: Not many middleware implement USAGEs. Submit a requirement to the component if you want.

# Certificate Chain Validation



Is the root CA in my key ring ?

Finish

Self signed:
Issuer=Subject

**Root CA**
Issuer – CN=Root CA,OU=Signers,O=IBM,C=US
Subject -CN=Root CA,OU=Signers,O=IBM,C=US
…
Signature

**Intermediate CA**
Issuer - CN=Root CA,OU=Signers,O=IBM,C=US
Subject – CN=Intermediate CA,OU=Signers,O=IBM,C=US
…
Signature

**End Entity**
Issuer – CN=Intermediate CA,OU=Signers,O=IBM,C=US
Subject -CN=Server Certificate,OU=z/OS,O=IBM,C=US
…
Signature

Start

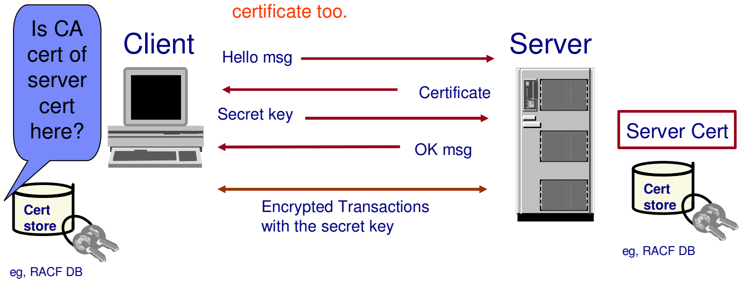
# Remember the simple rule – PVC

- **PVC - Parent Validates Child**
  - Child<-Parent<-Grandparent<-Great Grandparent<-…. <-Great Great….<-Root Grandparent
    - Ensure the content of the whole certificate chain has not been altered
      - Signature on the child verified by parent's public key
      - Signature on the root verified by its own public key
    - Trusting the Root Grandparent
      - Putting the root in the key store is the indication of trusting all its descendents

# Key ring plays a role in SSL handshake

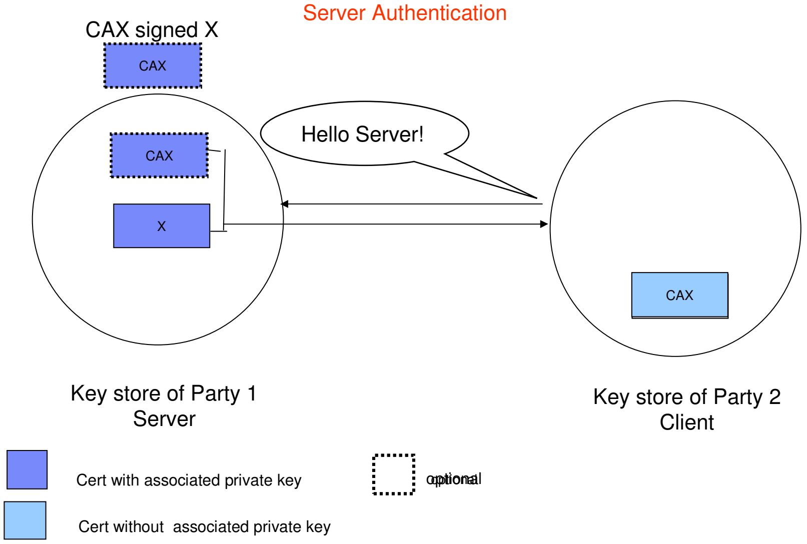
1. Client sends a ‘hello’ msg to server
2. Server sends its certificate to client
3. Client validates the server’s certificate
4. Client encrypts a secret key with server’s public key and sends it to server
5. Server decrypts the secret key with its private key
6. Server encrypts a ‘handshake OK’ msg with the secret key and sends it to client
7. Client trusts server, business can be conducted

* Note the above steps illustrate server authentication. For client authentication, server needs to validate client’s certificate too.



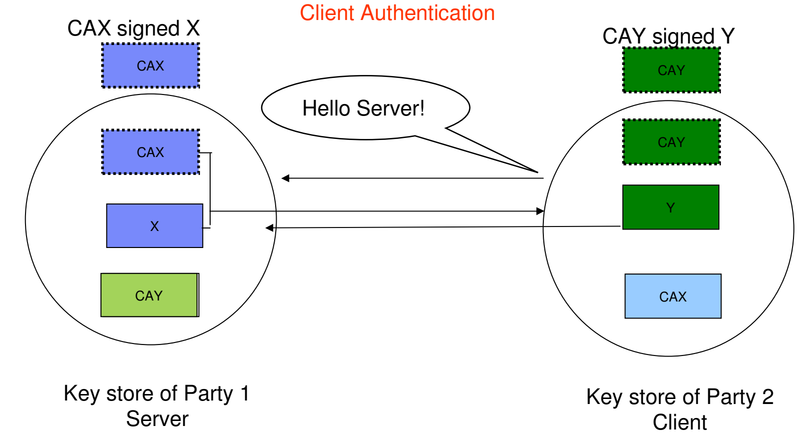
# Server Authentication

Server Authentication



# Client Authentication

Client Authentication



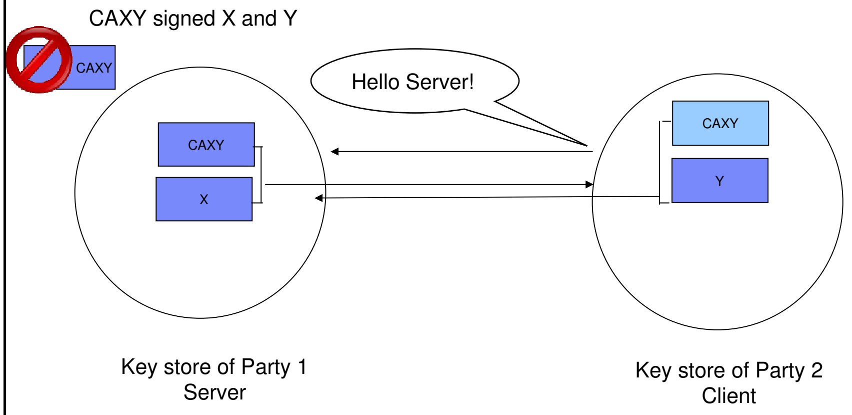
🟦 🟩 Cert with associated private key

⬚ optional

🟦 🟩 Cert without associated private key

# Simplify the set up for Client Authentication

Client Authentication



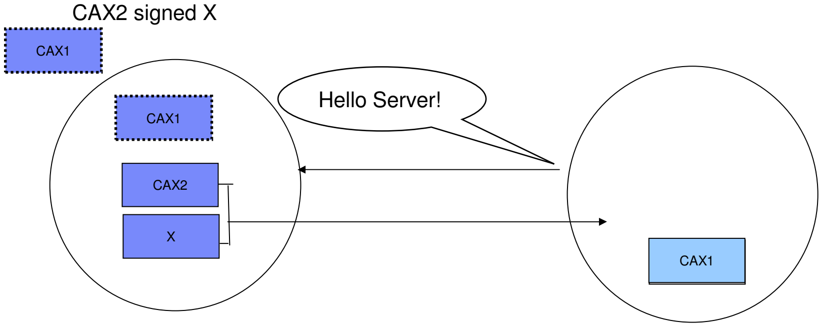
🟦 Cert with associated private key

🟦 Cert without associated private key

# Similar set up in the Chain scenario

CAX1 signed CAX2
CAX2 signed X

Server Authentication



Key store of Party 1
Server

Key store of Party 2
Client

■ Cert with associated private key

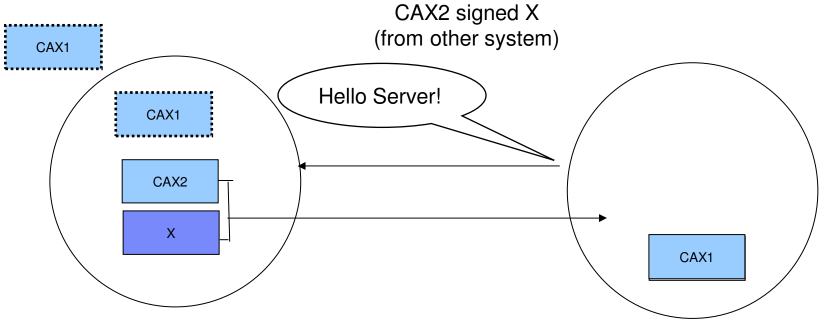
⬚ optional

□ Cert without associated private key

# Third Party CAs scenario

Server Authentication

CAX1 signed CAX2
CAX2 signed X
(from other system)



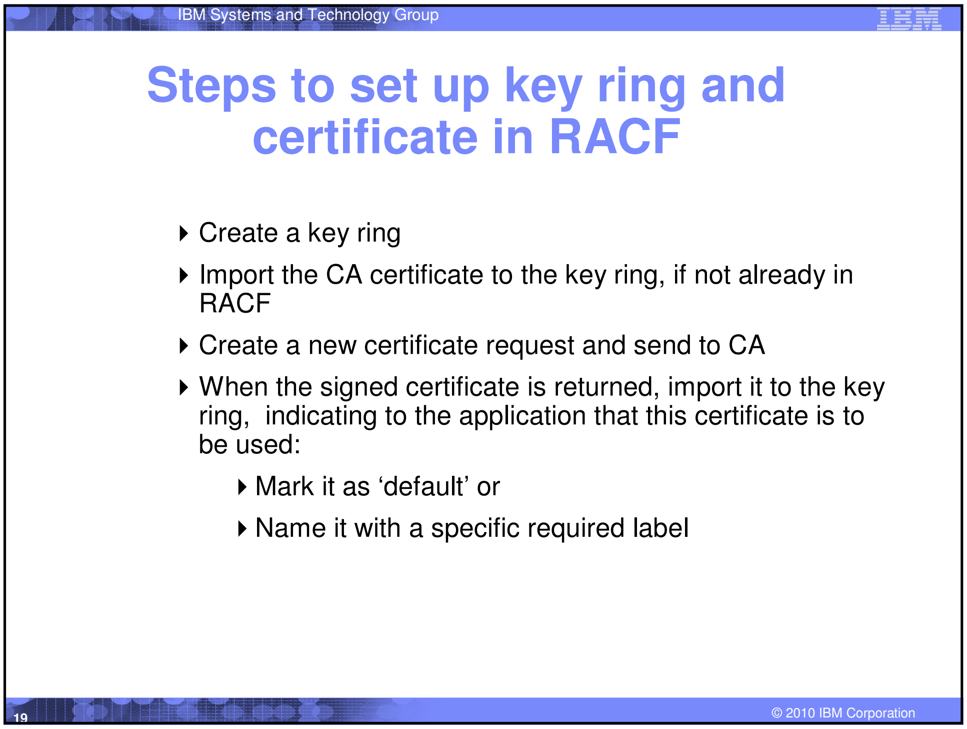
Key store of Party 1
Server

Key store of Party 2
Client

■ Cert with associated private key

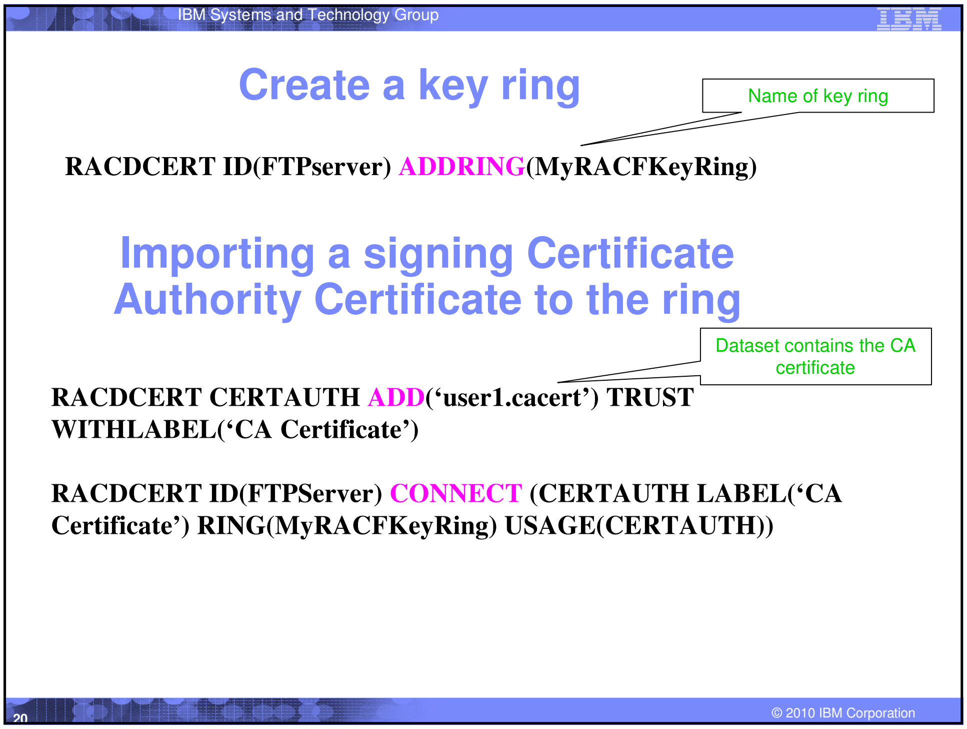
⬚ optional

□ Cert without associated private key



# Steps to set up key ring and certificate in RACF

- Create a key ring
- Import the CA certificate to the key ring, if not already in RACF
- Create a new certificate request and send to CA
- When the signed certificate is returned, import it to the key ring,  indicating to the application that this certificate is to be used:
  - Mark it as ‘default’ or
  - Name it with a specific required label



# Create a key ring

Name of key ring

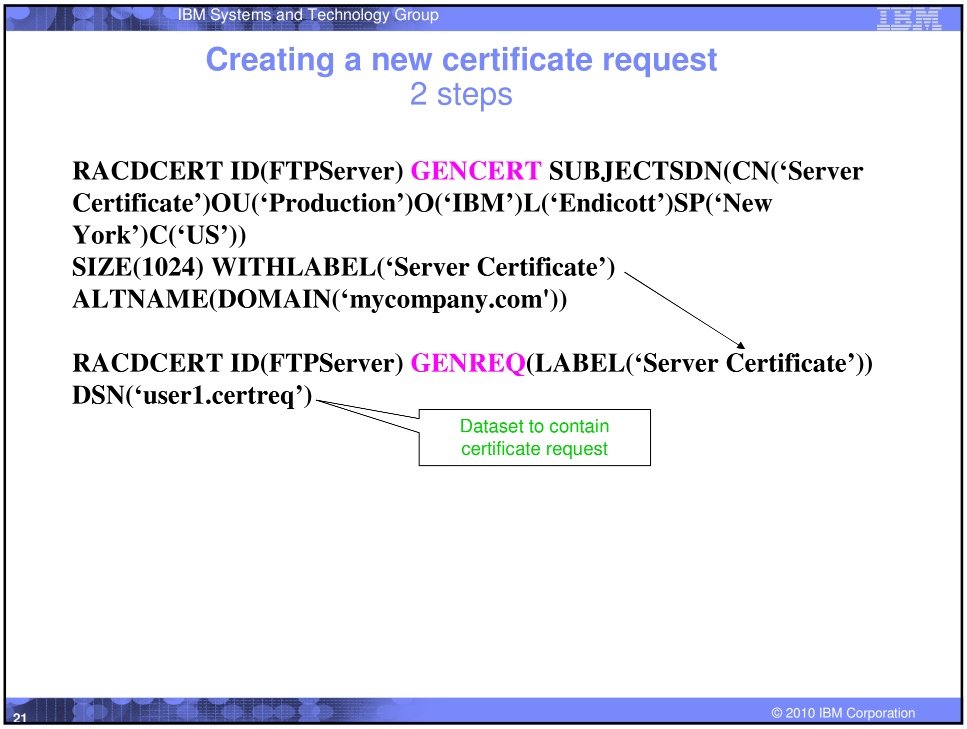
**RACDCERT ID(FTPserver) ADDRING(MyRACFKeyRing)**

# Importing a signing Certificate Authority Certificate to the ring

Dataset contains the CA certificate

**RACDCERT CERTAUTH ADD(‘user1.cacert’) TRUST WITHLABEL(‘CA Certificate’)**

**RACDCERT ID(FTPServer) CONNECT (CERTAUTH LABEL(‘CA Certificate’) RING(MyRACFKeyRing) USAGE(CERTAUTH))**

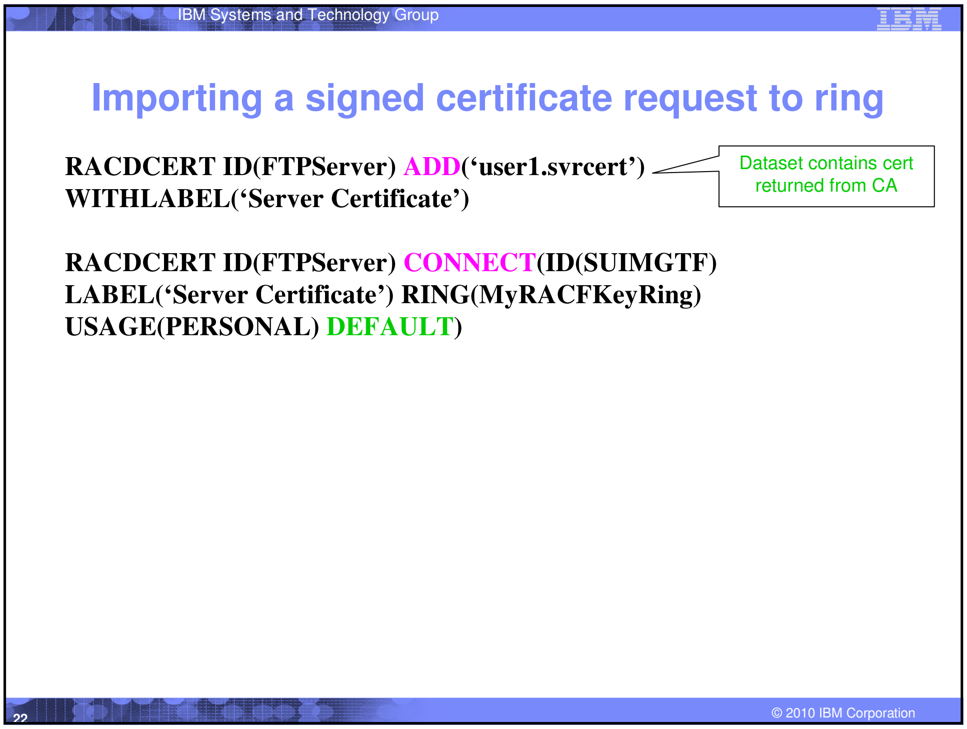


## Creating a new certificate request
### 2 steps

```
RACDCERT ID(FTPServer) GENCERT SUBJECTSDN(CN(‘Server
Certificate’)OU(‘Production’)O(‘IBM’)L(‘Endicott’)SP(‘New
York’)C(‘US’))
SIZE(1024) WITHLABEL(‘Server Certificate’)
ALTNAME(DOMAIN(‘mycompany.com'))

RACDCERT ID(FTPServer) GENREQ(LABEL(‘Server Certificate’))
DSN(‘user1.certreq’)
```

Dataset to contain certificate request

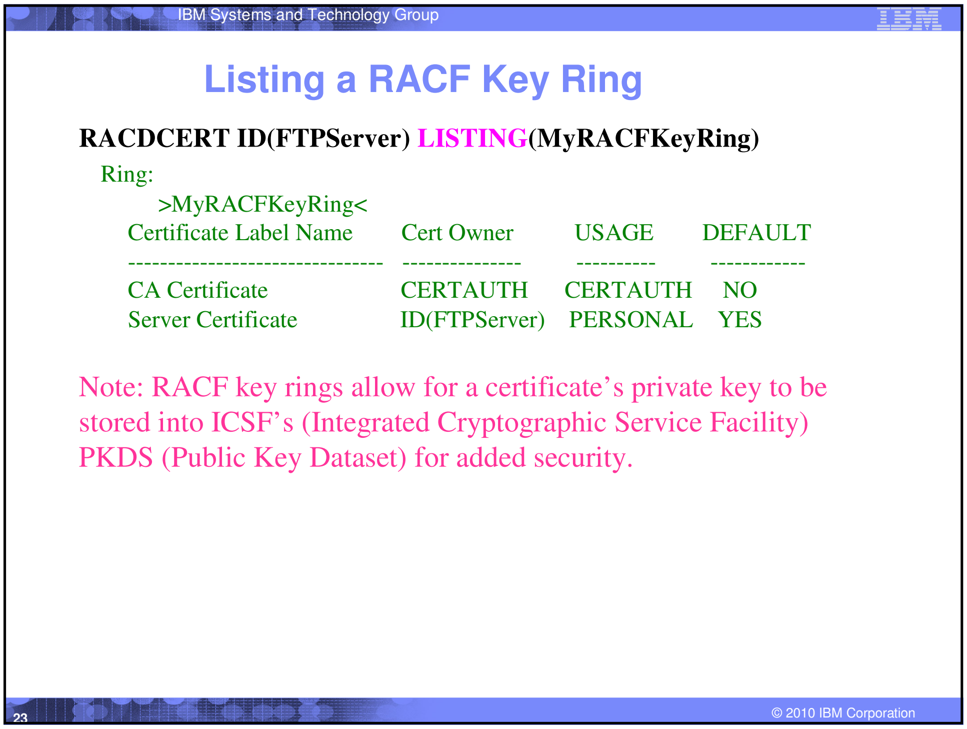


## Importing a signed certificate request to ring

```
RACDCERT ID(FTPServer) ADD(‘user1.svrcert’)
WITHLABEL(‘Server Certificate’)

RACDCERT ID(FTPServer) CONNECT(ID(SUIMGTF)
LABEL(‘Server Certificate’) RING(MyRACFKeyRing)
USAGE(PERSONAL) DEFAULT)
```

Dataset contains cert returned from CA

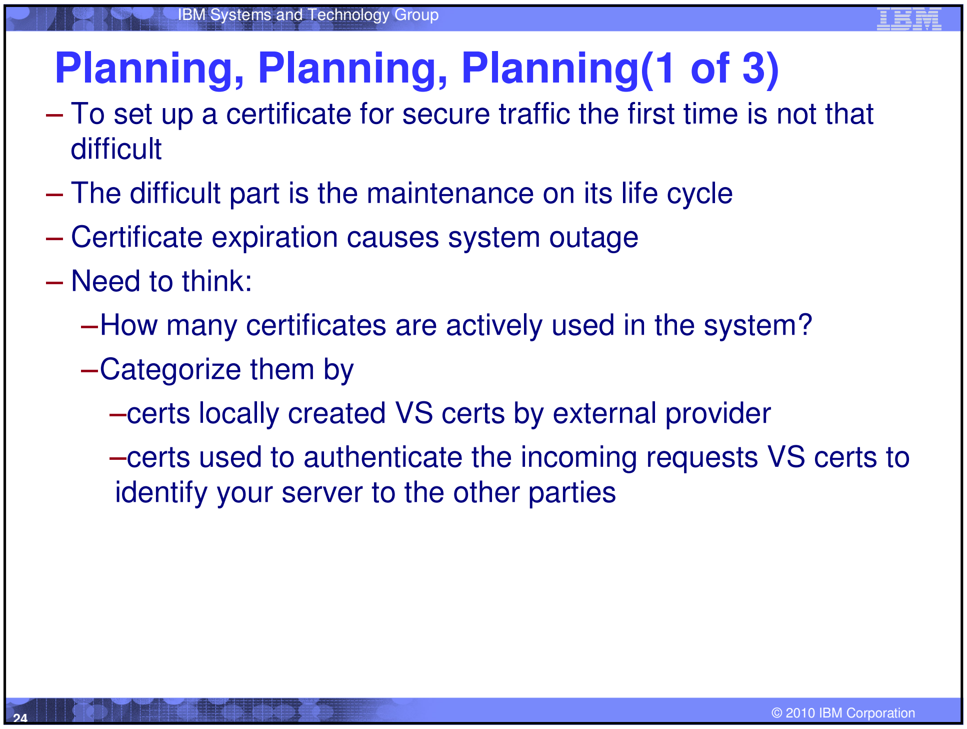


## Listing a RACF Key Ring

**RACDCERT ID(FTPServer) LISTING(MyRACFKeyRing)**

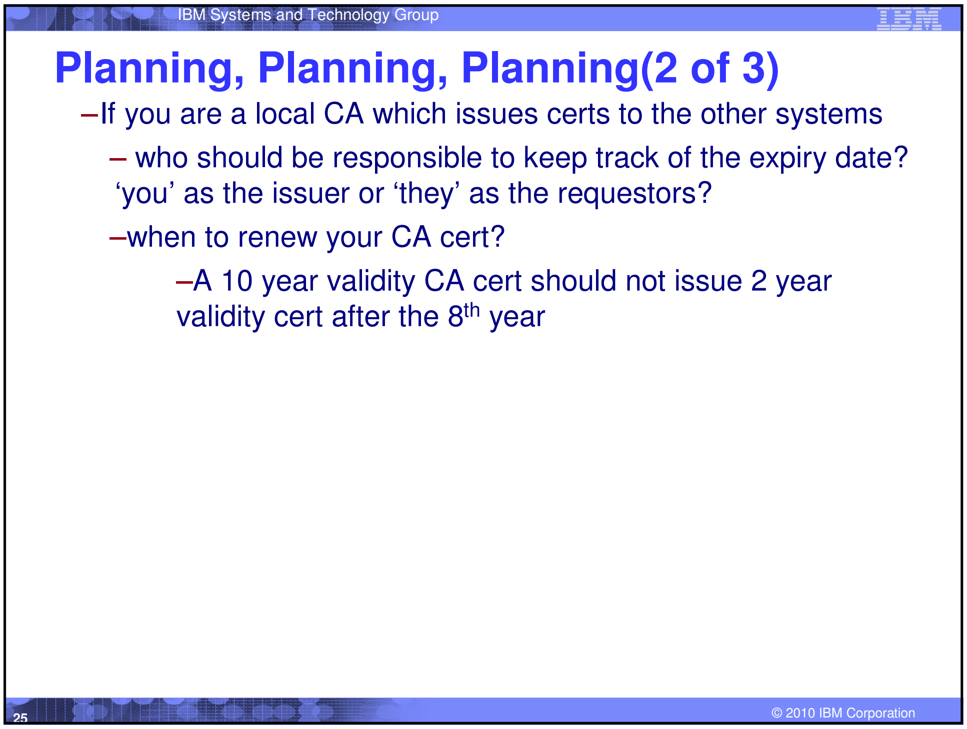
```
Ring:
     >MyRACFKeyRing<
   Certificate Label Name       Cert Owner       USAGE      DEFAULT
   --------------------------   --------------   ---------- ------------
   CA Certificate               CERTAUTH         CERTAUTH    NO
   Server Certificate           ID(FTPServer)    PERSONAL    YES
```

Note: RACF key rings allow for a certificate’s private key to be stored into ICSF’s (Integrated Cryptographic Service Facility) PKDS (Public Key Dataset) for added security.



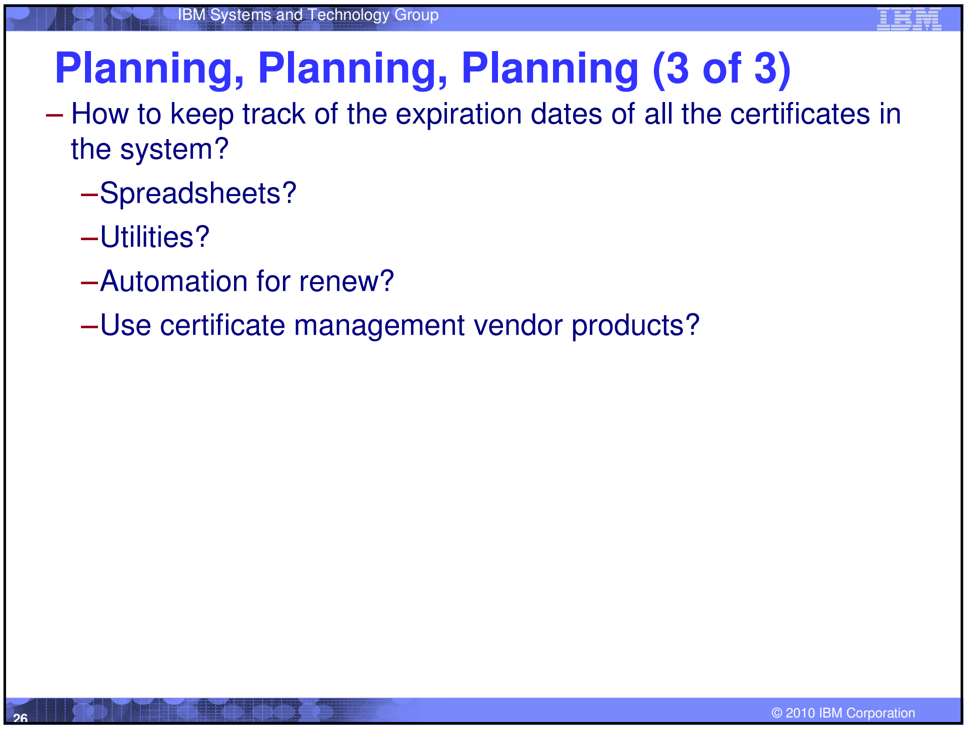
## Planning, Planning, Planning(1 of 3)

- To set up a certificate for secure traffic the first time is not that difficult
- The difficult part is the maintenance on its life cycle
- Certificate expiration causes system outage
- Need to think:
  - How many certificates are actively used in the system?
  - Categorize them by
    - certs locally created VS certs by external provider
    - certs used to authenticate the incoming requests VS certs to identify your server to the other parties



## Planning, Planning, Planning(2 of 3)

- If you are a local CA which issues certs to the other systems
  - who should be responsible to keep track of the expiry date? ‘you’ as the issuer or ‘they’ as the requestors?
  - when to renew your CA cert?
    - A 10 year validity CA cert should not issue 2 year validity cert after the 8th year

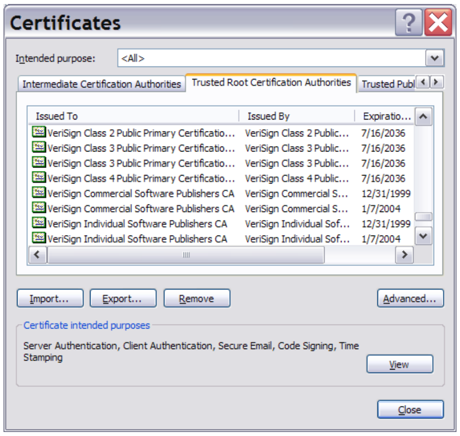


## Planning, Planning, Planning (3 of 3)

- How to keep track of the expiration dates of all the certificates in the system?
  - Spreadsheets?
  - Utilities?
  - Automation for renew?
  - Use certificate management vendor products?

# Build or Buy? (1 of 2)

- Who will be validating your certificate?
  - Global internet customers
    - Easier to buy from a well known CA since it is already installed in the browsers’ certificate store



Certificates

Intended purpose: <All>

Intermediate Certification Authorities
Trusted Root Certification Authorities
Trusted Publ

| Issued To | Issued By | Expiratio... |
|---|---|---|
| VeriSign Class 2 Public Primary Certificatio... | VeriSign Class 2 Public... | 7/16/2036 |
| VeriSign Class 3 Public Primary Certificatio... | VeriSign Class 3 Public... | 7/16/2036 |
| VeriSign Class 3 Public Primary Certificatio... | VeriSign Class 3 Public... | 7/16/2036 |
| VeriSign Class 4 Public Primary Certificatio... | VeriSign Class 4 Public... | 7/16/2036 |
| VeriSign Commercial Software Publishers CA | VeriSign Commercial S... | 12/31/1999 |
| VeriSign Commercial Software Publishers CA | VeriSign Commercial S... | 1/7/2004 |
| VeriSign Individual Software Publishers CA | VeriSign Individual Sof... | 12/31/1999 |
| VeriSign Individual Software Publishers CA | VeriSign Individual Sof... | 1/7/2004 |

Certificate intended purposes

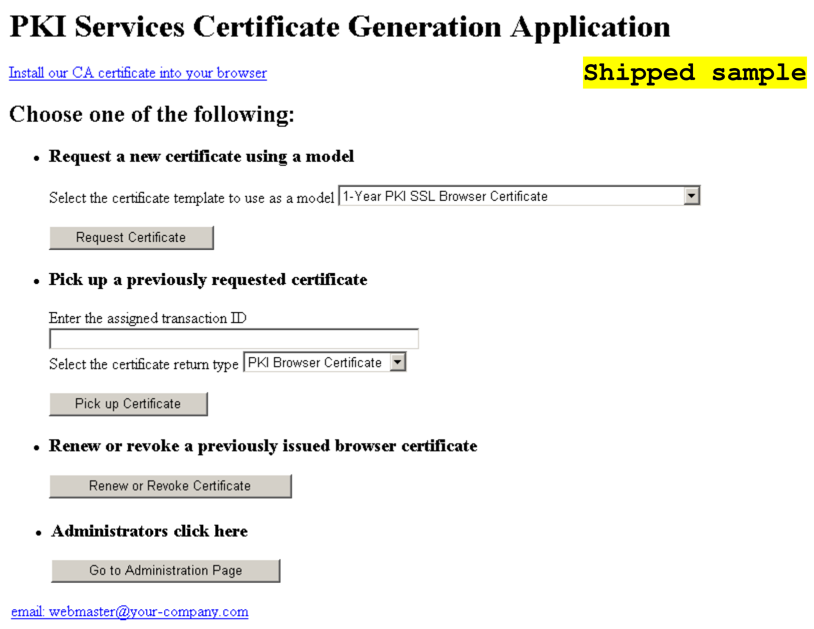
Server Authentication, Client Authentication, Secure Email, Code Signing, Time Stamping

# Build or Buy?(2 of 2)

- Internal servers, employees
  - Build your own since you can have the internal CA certs distributed easily
- Business partners
  - Either way
  - If you already built a trust relationship with the partners, there should be no problem for them to install your CA cert

# Certificate Authority on z/OS

- PKI Services provides full certificate life cycle management
  - Request, create, renew, revoke certificate
  - Provide certificate status through Certificate Revocation List(CRL) and Online Certificate Status Protocol (OCSP)
  - Generation and administration of certificates via customizable web pages
  - Support Simple Certificate Enrollment Protocol (SCEP) for routers to request certificates automatically
- Not a priced product. Licensed with z/OS
  - Cost efficient for banks, government agencies to host Digital Certificate management
- Provide more functions than RACDCERT
- Provide expiration notification and automatic renewal



## PKI Services Certificate Generation Application

Install our CA certificate into your browser

Shipped sample

### Choose one of the following:

- **Request a new certificate using a model**

  Select the certificate template to use as a model: 1-Year PKI SSL Browser Certificate

- **Pick up a previously requested certificate**

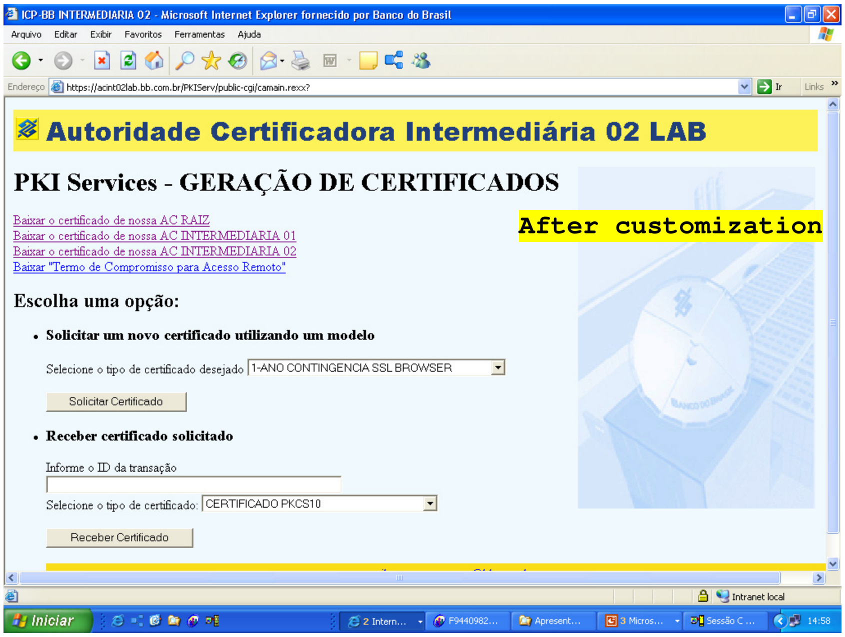
  Enter the assigned transaction ID

  Select the certificate return type: PKI Browser Certificate

- **Renew or revoke a previously issued browser certificate**

- **Administrators click here**

email: webmaster@your-company.com



Endereço: https://acint02lab.bb.com.br/PKIServ/public-cgi/camain.rexx?

**Autoridade Certificadora Intermediária 02 LAB**

**PKI Services - GERAÇÃO DE CERTIFICADOS**

Baixar o certificado de nossa AC RAIZ
Baixar o certificado de nossa AC INTERMEDIARIA 01
Baixar o certificado de nossa AC INTERMEDIARIA 02
Baixar "Termo de Compromisso para Acesso Remoto"

**Escolha uma opção:**

- **Solicitar um novo certificado utilizando um modelo**

  Selecione o tipo de certificado desejado: 1-ANO CONTINGENCIA SSL BROWSER

  Solicitar Certificado

- **Receber certificado solicitado**

  Informe o ID da transação

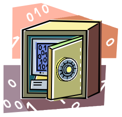
  Selecione o tipo de certificado: CERTIFICADO PKCS10

  Receber Certificado

**After customization**

# Other benefits of using PKI Services

- Provide options for requestor to generate his own key pa request the PKI CA to generate it
- Relatively low MIPS to drive thousands of certificates
- Leverage existing z/OS skills and resources
- Run in separate z/OS partitions (integrity of zSeries® LPARs)
- Scalable  (Sysplex exploitation)



# Major Prerequisite Products

## –RACF (or equivalent)

- For storing PKI CA certificate
- For authorization

## –IBM z/OS HTTP Server / Websphere Application Server

- For web page interface

## –LDAP Directory (z/OS or other platforms)

- For publishing issued certificates and CRLs
- For email notification

## –ICSF (optional)

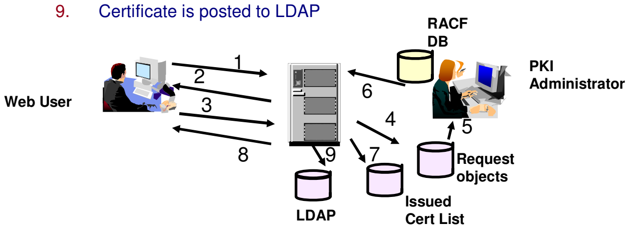
- For more secure CA private key
- For PKI CA to generate key pair

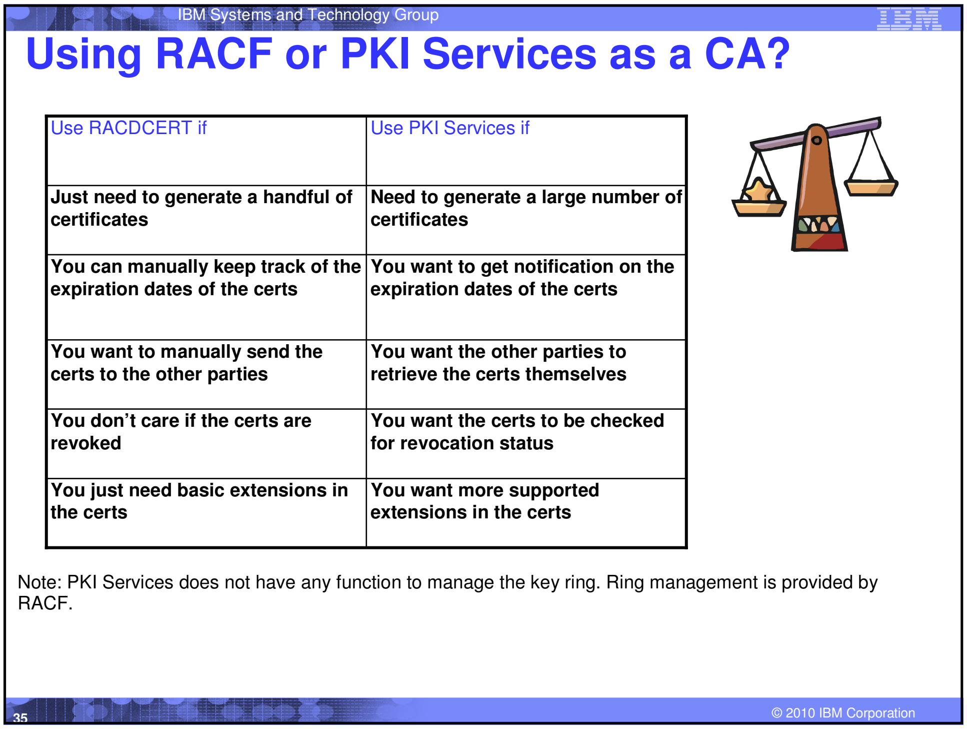
## –z/OS Communications Server (optional)

- For email notification

# z/OS PKI Services Process Flow – a simplified sample view

1. User contacts PKI Services to request for certificate
2. CGI/JSP constructs a web page for user to input information
3. CGI/JSP packages all the info and send to the callable service
4. Callable service calls the daemon to generate the request object and put it in the Request objects DB
5. Administrator approves the request through the administrator web page
6. CGI/JSP calls callable service which in turn calls the daemon to create the certificate, sign with the CA key in the RACF DB
7. Certificate is placed in the Issued Cert List DB
8. Certificate is sent to the user
9. Certificate is posted to LDAP

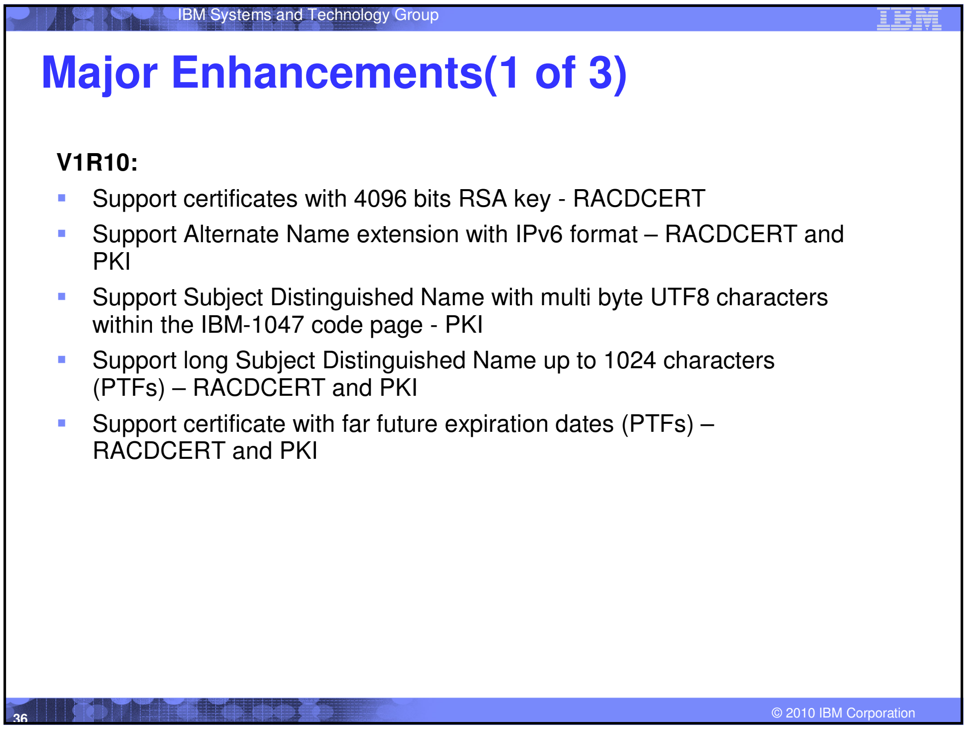




# Using RACF or PKI Services as a CA?

| Use RACDCERT if | Use PKI Services if |
|---|---|
| **Just need to generate a handful of certificates** | **Need to generate a large number of certificates** |
| **You can manually keep track of the expiration dates of the certs** | **You want to get notification on the expiration dates of the certs** |
| **You want to manually send the certs to the other parties** | **You want the other parties to retrieve the certs themselves** |
| **You don't care if the certs are revoked** | **You want the certs to be checked for revocation status** |
| **You just need basic extensions in the certs** | **You want more supported extensions in the certs** |

Note: PKI Services does not have any function to manage the key ring. Ring management is provided by RACF.



# Major Enhancements(1 of 3)

**V1R10:**

- Support certificates with 4096 bits RSA key - RACDCERT
- Support Alternate Name extension with IPv6 format – RACDCERT and PKI
- Support Subject Distinguished Name with multi byte UTF8 characters within the IBM-1047 code page - PKI
- Support long Subject Distinguished Name up to 1024 characters (PTFs) – RACDCERT and PKI
- Support certificate with far future expiration dates (PTFs) – RACDCERT and PKI

# Major Enhancements(2 of 3)

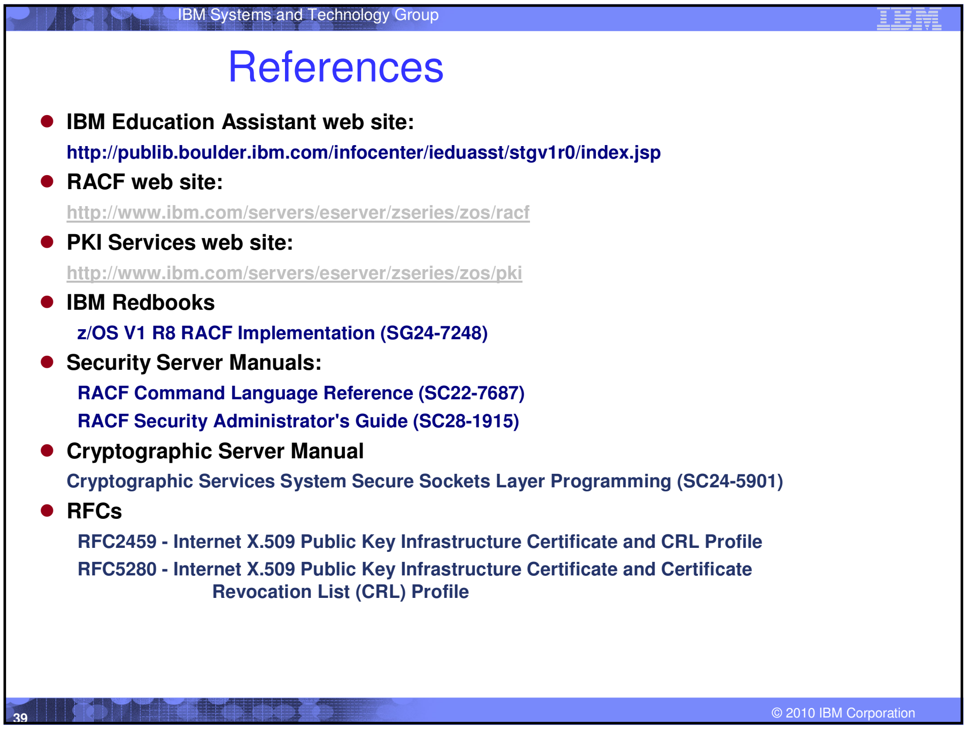
**V1R11:**

- Generate certificate with supplied public key that was already stored in PKDS - RACDCERT
- Support certificate with multi byte UTF8 characters in the subject distinguished name, even they are outside the IBM-1047 code page - RACDCERT
- Provide option for the user to request PKI CA to generate the key pair - PKI
- Support SHA256 in the signing algorithm - PKI
- Implement the web pages with XML and JSPs to facilitate the integration with PKI Services from other applications - PKI

# Major Enhancements (3 of 3)

**V1R12 (Preview)**

- Support Elliptic Cryptographic Curve (ECC) keys – RACDCERT and PKI
- Support Certificate Management Protocol (CMP) clients to communicate with PKI Services – PKI
- Support the creation of custom extensions to certificate - PKI



# References

- **IBM Education Assistant web site:**

  **http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp**

- **RACF web site:**

  http://www.ibm.com/servers/eserver/zseries/zos/racf

- **PKI Services web site:**

  http://www.ibm.com/servers/eserver/zseries/zos/pki

- **IBM Redbooks**

  **z/OS V1 R8 RACF Implementation (SG24-7248)**

- **Security Server Manuals:**

  **RACF Command Language Reference (SC22-7687)**

  **RACF Security Administrator's Guide (SC28-1915)**

- **Cryptographic Server Manual**

  **Cryptographic Services System Secure Sockets Layer Programming (SC24-5901)**

- **RFCs**

  **RFC2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile**

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