

The Digital Certificate Journey from RACF to PKI Services

SHARE New York, NY
Session 1791
August 19th 2004

Wai Choi
IBM Corporation
Poughkeepsie, NY

Phone: (845) 435-7623
e-mail: wchoi@us.ibm.com



Agenda

- **PKI Basics**
- **RACF Support for Certificates**
- **PKI Services Support for Certificates**
- **What's new on PKI Services for z/OS Release 5**

What is PKI?

- **Public Key Infrastructure based on the public key cryptography to create, manage, store, distribute, verify digital certificates**
- **Not like the secret key cryptography which encrypts and decrypts with the same key**
- **Involves a public-private key pair, encrypts with one key and decrypts with its partner key**
- **To facilitate 2 main goals of secure communication – confidentiality and integrity**

Encryption (for confidentiality) Under Public Key Cryptography

Encrypting a message:



Decrypting a message:



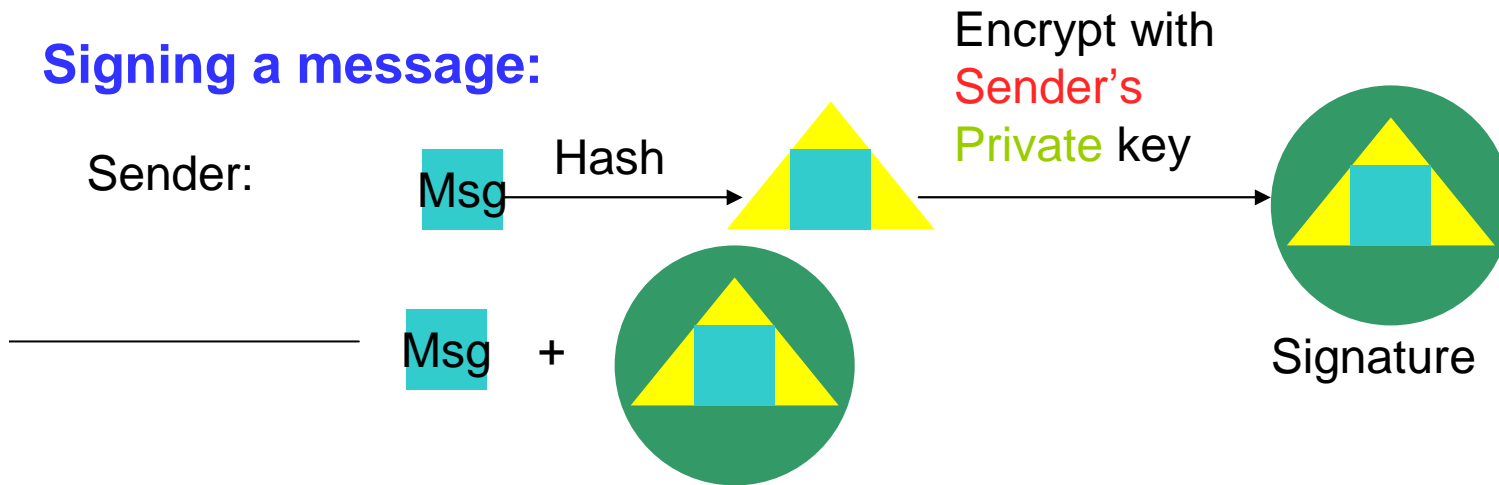
Keys:

 Plain text

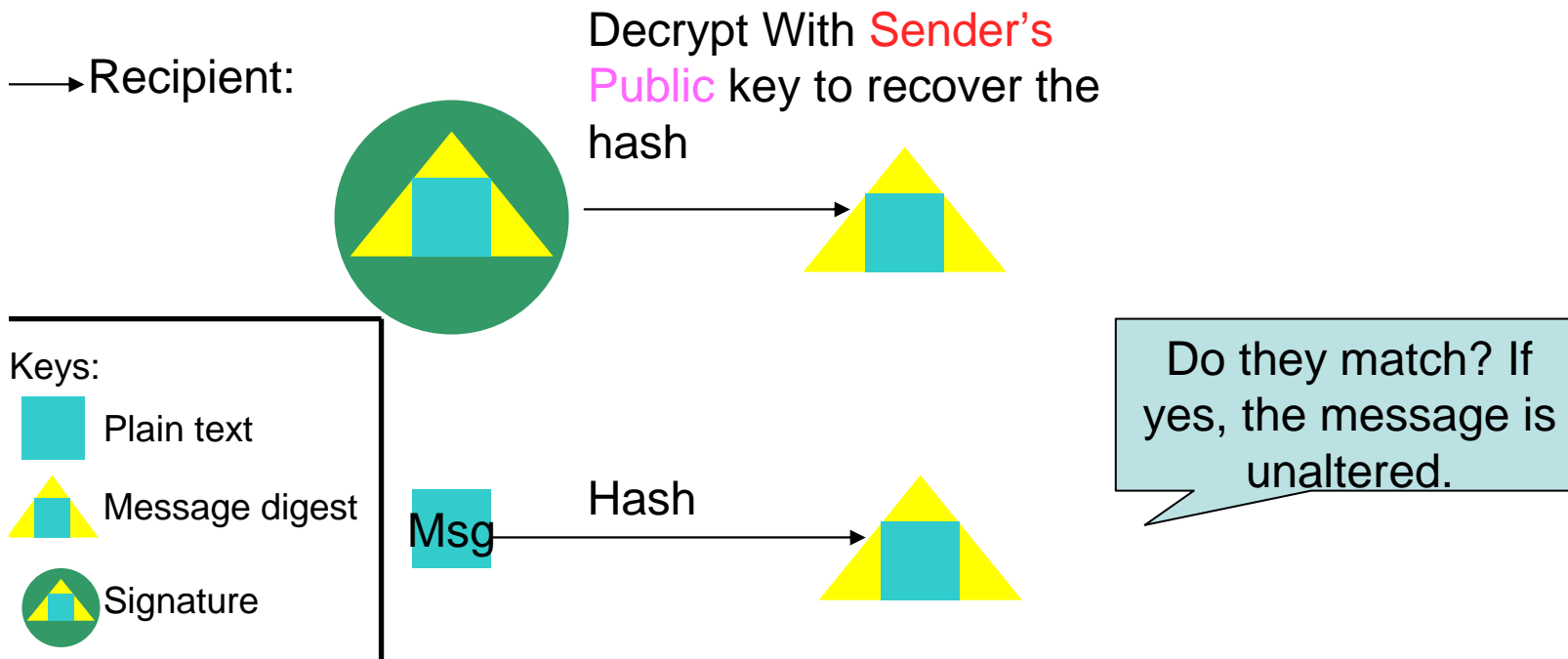
 Encrypted text

Signing (for integrity) Under Public Key Cryptography

Signing a message:



Verifying a message:



Public key and Certificate

- Public key alone can not tell who the key owner is
- Need a trusted third party, Certificate Authority (CA), to bind a public key to a subject through a certificate
- The Certificate Authority signs the certificate with its private key to prove its authenticity

What's inside a Certificate?

Certificate Info

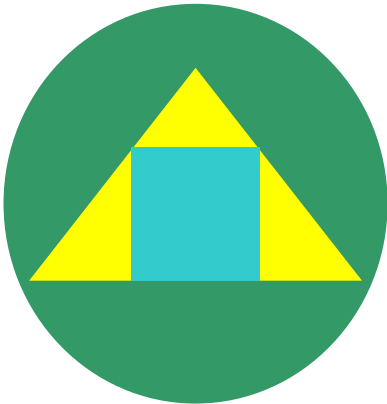
version
serial number
signature algorithm ID
issuer's name
validity period
subject's name
subject's public key
extensions

This is the hash/encrypt algorithm used in the signature

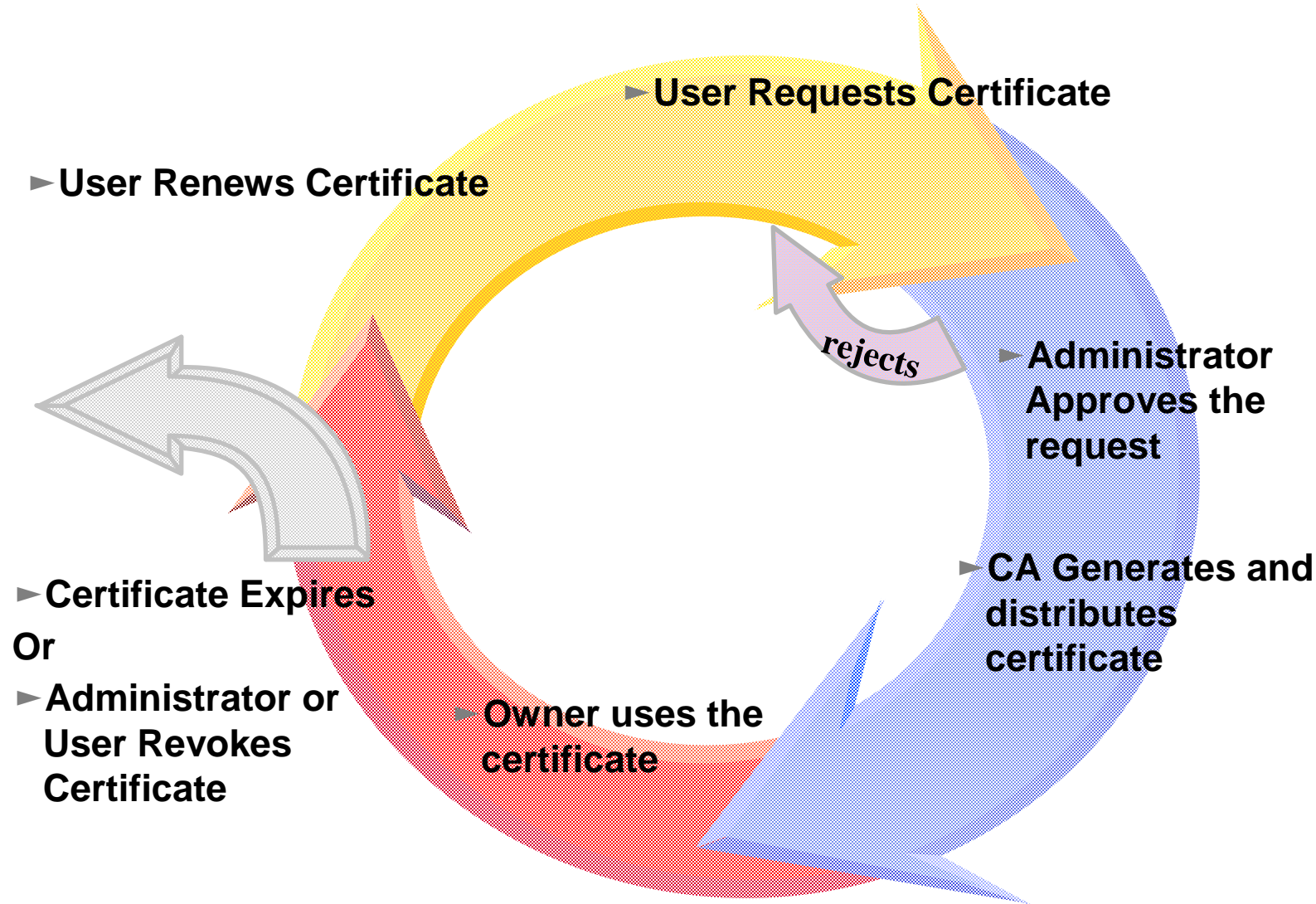
The certificate binds a public key to a subject

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

Certificate Signature



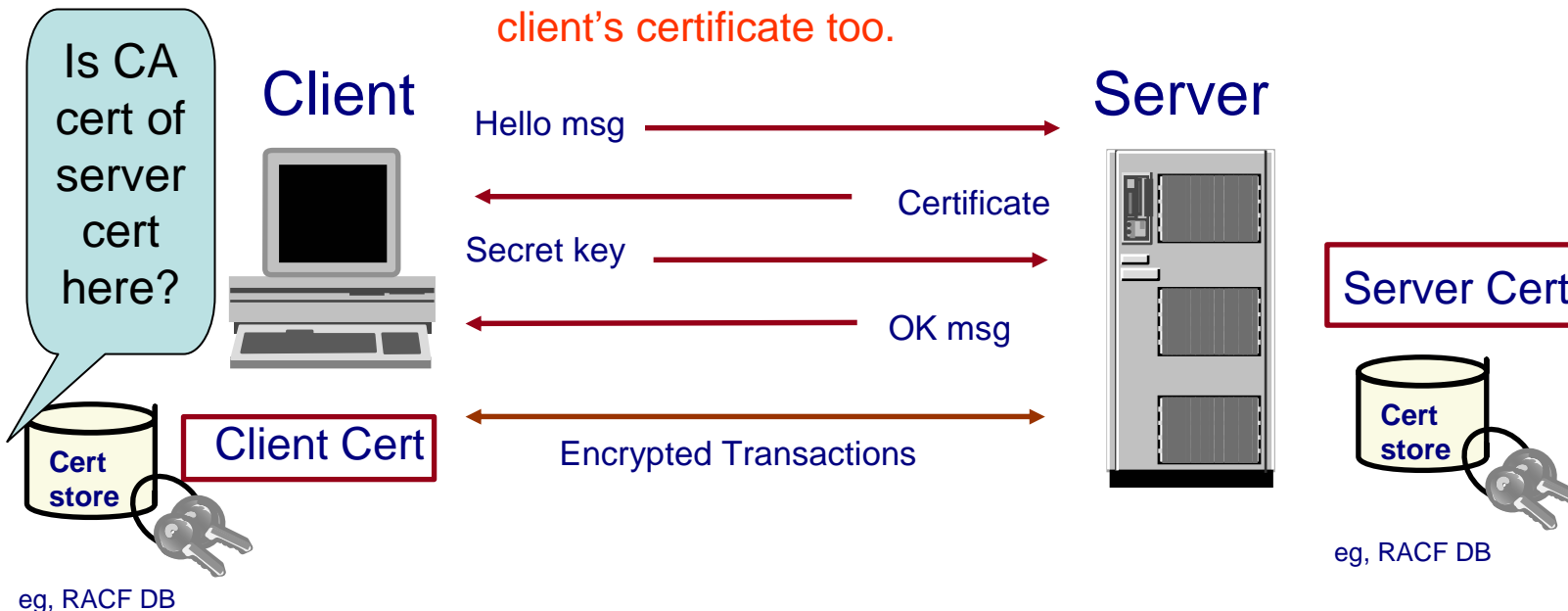
Certificate Life Cycle – This is why you need PKI



A common use of Certificate - 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.



eg, RACF DB

eg, RACF DB

Two Basic PKI Operations

Certificate generation (In response to a user request)

- Both RACF and PKI Services can be used as a Certificate Authority

Certificate validation

involves the questions of:

- Whether you **trust** the issuer of the certificate – is it in your certificate store, key ring...
- Whether the certificate has a valid **signature** of the issuer
- Whether the certificate is **expired**
- Whether the certificate has been **revoked** (see slide 45)
 - PKI Services supports Certificate Revocation Lists (CRLs), RACF doesn't
- Whether the certificate contains **information that is specific** to your application that uses that certificate. This includes specific extensions that your application is looking for.

RACF support for Certificates

1. Certificate generation – RACDCERT GENCERT, GENREQ (see slide 46)

● RACDCERT GENCERT:

- **GENerates a CERTificate, and optionally public-private key**
- **Can utilize z/OS hardware crypto for private key generation, storage and operation**
- **Certificate can be self signed or signed by another certificate's corresponding private key**
- **May create certificates for other servers, includes those on non-z/OS platform**

RACF support for Certificates...

● RACDCERT GENREQ:

- GENerates a REQuest by copying information, including the public key, from a previously created certificate
- Usually issued after GENCERT a self signed certificate
- The request is signed with the private key associated with the specified certificate
- The request is saved to a data set in the Base64 format which can be used in cut and paste (see slide 49)
- The created certificate request can be submitted (e.g. e-mail it, paste it into a web page, etc) to a CA for issuance
- Note there is no key generation in GENREQ

RACF support for Certificates...

2. Certificate installation – RACDCERT ADD, ADDRING, CONNECT (see slides 47, 48)

■ RACDCERT ADD:

- Install a certificate to RACF with or without private key

■ RACDCERT ADDRING

- Create a key ring (for handshake process)
- Certificate must be placed in a key ring before it can be used by other middleware, eg. SSL

■ RACDCERT CONNECT

- Place a certificate in a key ring
- You may place your own certificates and other's certificate in your key ring
- A certificate can be placed in more than one key ring, even with different usages – PERSONAL, CERTAUTH and SITE
- The TRUST status of the connected certificate in the ring tells if the certificate can be used

RACF support for Certificates...

3. Certificate administration – RACDCERT LIST, ALTER, DELETE, REMOVE, ...

● RACDCERT LIST:

- Display certificate information for a userid

● RACDCERT ALTER

- Change the TRUST/NOTRUST status or the label of a certificate

● RACDCERT DELETE

- Remove a certificate from RACF

● RACDCERT REMOVE

- Remove a certificate from a key ring

● And more...

Another common use of Certificate

– Log in a system without userid/password

- Handshake process requires one's own certificate and the issuer's certificate of the communicating party in each other's certificate store/key ring in the system.
- The system in which the user logs in needs to have the user's certificate installed.
- It will be a huge administrative burden to add and manage thousands of user certificates for all the users who use certificate to log in.

More Sophisticated Support from RACF

- **Solution 1: Certificate Name Filtering**
 - Supported by RACF
- **Solution 2: HostIdMapping**
 - Supported by RACF and PKI Services
- **No user certificate needs to be installed.**

More Sophisticated Support from RACF...

- **Certificate Name Filtering – RACDCERT MAP** (see slide 48)
 - Create the definition of a set of rules ('filters') based on the subject's or issuer's distinguished names (or both)
 - Can map one or more certificates to a filter
- **HostIdMapping**
 - A client can present a certificate containing a HostIdMapping extension to the server
 - This extension contains a host name and a subject id
 - RACF will honor this extension if
 - the issuing CA cert is marked HIGHTRUST
 - the host name in the extension matches a profile IRR.HOST.<host name> in the SERVAUTH class
 - PKI Services can create this extension

Introduction to PKI Services

- **New component on z/OS since V1R3**
- **Closely tied to RACF, but supports more functions than RACDCERT**
- **Complete Certificate Authority /Registration Authority (CA/RA) package**
 - **Full certificate life cycle management: request, create, renew, revoke**
- **Generation and administration of certificates via customizable web pages**
- **Support automatic or administrator approval process**
- **Create Certificate Revocation Lists (CRLs)**
- **Certificates and CRLs can be posted to LDAP**
- **Provides email notification for completed certificate request and expiration warnings**

Introduction to PKI Services...

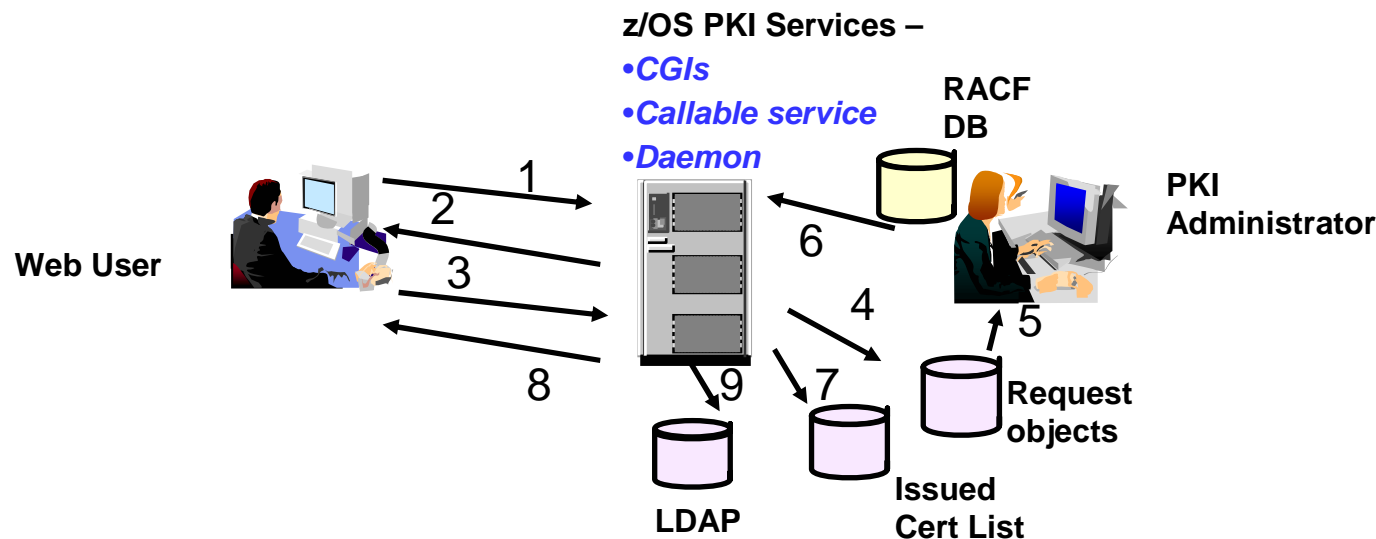
- **Provides Trust Policy Plug-in for certificate validation**
- **Manual - "PKI Services Guide and Reference"**

Benefits of using PKI Services on z/OS

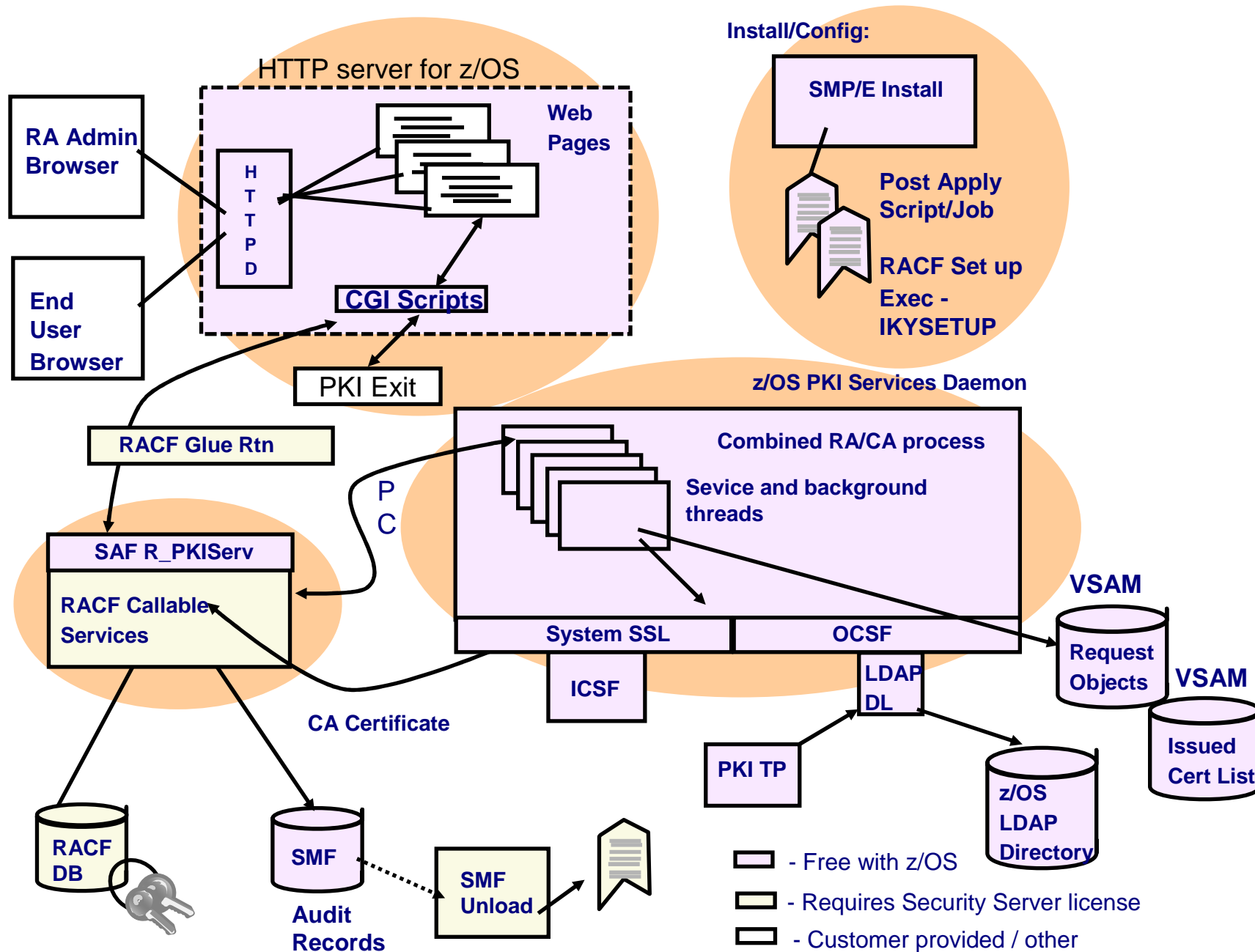
- No additional cost with z/OS, no need to pay a third party CA for certificates
- Relatively low mips to drive thousands of certificates
- Leverage existing z/OS skills and resources
- Ability to host Digital Certificate management for the banks, government agencies...
- Run independently of other workloads
- Run in separate z/OS partitions (integrity of zSeries LPARs)
- Scalable (Sysplex exploitation)
- Secure with zSeries cryptography

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

1. User contacts PKI Services to request for certificate
2. CGI constructs a web page for user to input information
3. CGI 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 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



z/OS PKI Services Architecture

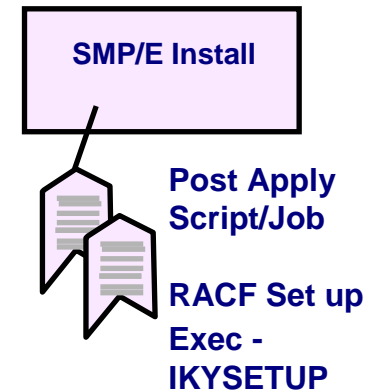


z/OS PKI Services Architecture...

● IKYSETUP

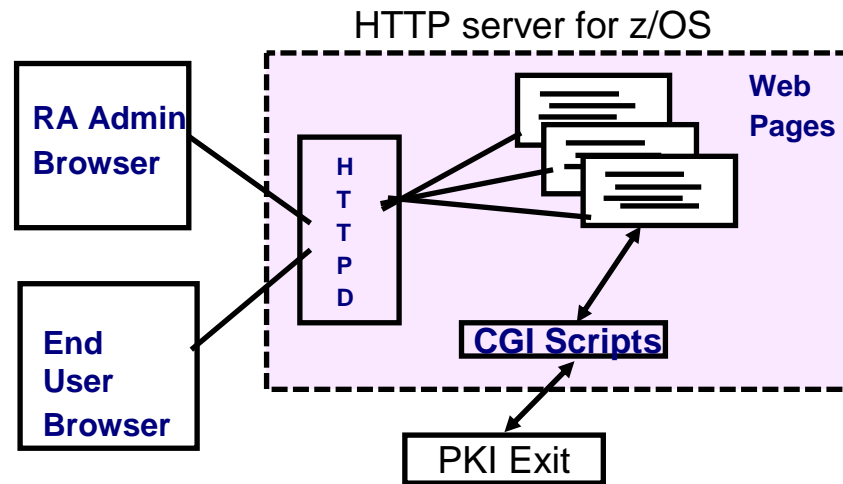
- A REXX exec shipped in SYS1.SAMPLIB to perform RACF administration tasks for setting up PKI Services:
 - Add administration groups, daemon user IDs, client surrogate IDs
 - Set up access control profiles
 - Create or install PKI Services CA certificate and Web Server certificate for the PKI web pages

Install/Config:



z/OS PKI Services Architecture...

● Browser/CGI interface



- Web page contents are defined in a certificate template file, pkiserv.tmpl (see slide 50)
- The CGI tasks include:
 - read the template file to form the web pages
 - package up all the input values from the web page and constant values from the template file as parameters to call the R_PKIServ callable service
 - provide hooks to call installation-provided exit routine for customization

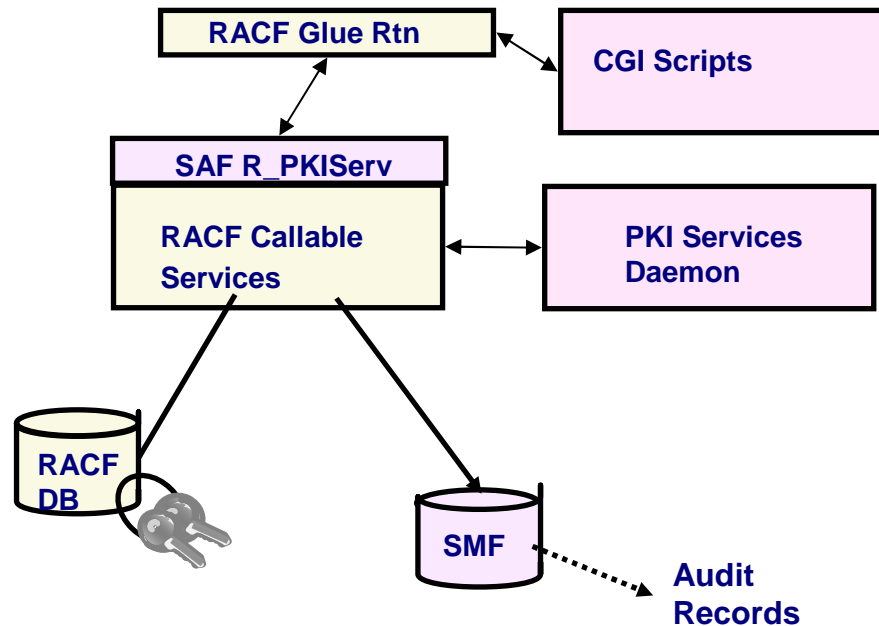
z/OS PKI Services Architecture...

● Browser/CGI interface...

- You may request a browser certificate or a server cert
 - Browser certificate
 - The public-private key pair is generated by the browser
 - The browser will then create a certificate request signed with the private key
 - The returned certificate can be installed directly into the browser
 - Server certificate
 - Similar process as in browser certificate generation except that the public-private key pair is generated on the server side
- Note PKI Services is not involved in generating key pairs in browser nor server certificate generation.

z/OS PKI Services Architecture...

● SAF callable service – R_PKIServ



- Interface between CGIs and the PKI Services Daemon (through the glue routine)
- Performs authorization checking and parameter validation
- Provides functions for end user and administrator

z/OS PKI Services Architecture...

● SAF callable service – R_PKIServ...

—End user functions are acting on certificates:

- Request
- Export
- Verify
- Renew
- Suspend
- Revoke

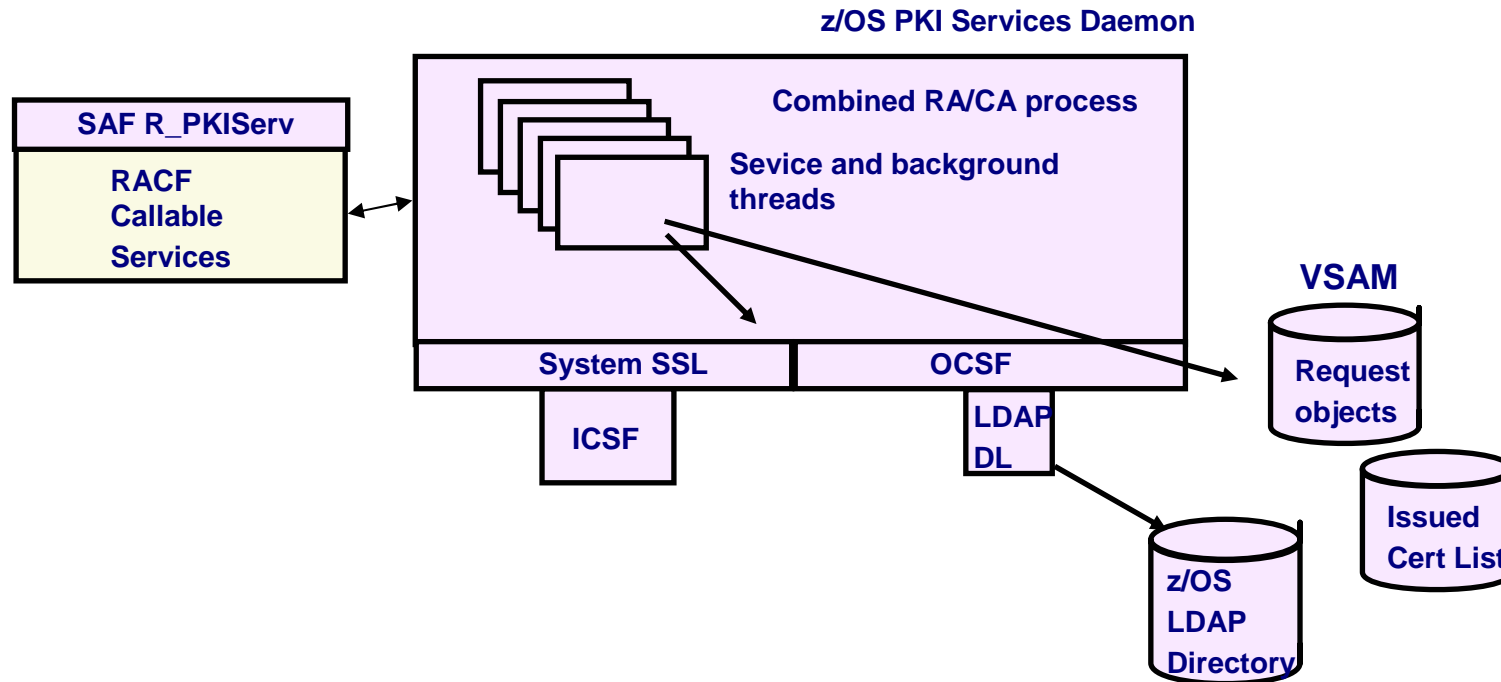
z/OS PKI Services Architecture...

● SAF callable service – R_PKIServ...

- Administrator functions are acting on certificate requests and certificates:
 - Query (request, certificate)
 - Approve (request)
 - Modify (request)
 - Reject (request)
 - Suspend (certificate)
 - Resume (certificate)
 - Revoke (certificate)

z/OS PKI Services Architecture...

● PKI Services Daemon



- Invoked by the R_PKIServ callable service
- Perform the real work
- Read the configuration file, pkiserv.conf, to determine the set up values (see slide 51)
- VSAM datasets are used to store request objects (object store) and Issued Certificates List (ICL)

Ready, Set, Go – Be your own CA Issuing Certificates for your customers

1. Generate a self signed PKI Services CA certificate
 - `RACDCERT CERTAUTH GENCERT SUBJECT(...)
WITHLABEL('PKI CA cert')...`
2. Create a key ring called pki_ca_ring for PKI Services,
under ID PKISRVDD
 - `RACDCERT ID(PKISRVDD) ADDRING(pki_ca_ring)`
3. Connect the CA cert to the ring
 - `RACDCERT ID(PKISRVDD) CONNECT(CERTAUTH LABEL('PKI
CA cert') RING(pki_ca_ring) USAGE(PERSONAL)
DEFAULT)`
4. Specify the key ring in the PKI Services configuration file,
pkiserv.conf
 - `KeyRing=PKISRVDD/pki_ca_ring`

Ready, Set, Go – Be your own CA Issuing Certificates for your customers...

5. Customize other values in the configuration file and in the template file, pkiserv.tmpl
6. Set up the web server --create certificate, key ring...
7. Start PKI Services and web server
8. Ready for a customer to request a browser certificate through the PKI Services web page

PKI Services Certificate Generation Application

[Install our CA certificate into your browser](#)

Choose one of the following:

This is the PKI Services home page

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

Select the certificate template to use as a model

Request Certificate

Pull-down for requesting a certificate of a certain type (template)

- **Pick up a previously requested certificate**

Enter the assigned transaction ID

Select the certificate return type

Pick up Certificate

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

Renew or Revoke Certificate

- **Administrators click here**

Go to Administration Page

Link for renew/revoke forces SSL client authentication

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Admin link is either userid/pw protected or forces SSL client authentication

1-Year SSL Browser Certificate

Choose one of the following:

- **Request a New Certificate**

Enter values for the following field(s)

Your name for tracking this request (optional)

Email address for distinguished name (optional)

Common Name

Email address for notification purposes (optional)

Pass phrase for securing this request. You will need to supply this value when retrieving your certificate

Reenter your pass phrase to confirm

Select the following key information

Cryptographic Service Provider

Enable strong private key protection?

- **Pick Up a Previously Issued Certificate**

This is the certificate request page. The input boxes that appear on this page depends on the <CONTENT> section of the certificate template chosen (See slide 50)

When the submit button is pressed, the data entered by the user and the data hardcoded for this certificate template under the <CONSTANT> section is sent to PKI Services for processing. (See slide 50)

email_webmaster@your-company.com

Request submitted successfully

Here's your transaction ID. You will need it to retrieve your certificate. Press 'Continue' to retrieve the certificate.

1jSuReRBV2VS2SHV+++++++

Continue

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

The request is queued to PKI Services request database for approval. The result is the return of a transaction ID

PKI Services Administration

Choose one of the following:

- Work with a single certificate request

Enter the Transaction ID:

Process Request

- Work with a single issued certificate

Enter the Serial Number:

Process Certificate

- Specify search criteria for certificates and certificate requests

Certificate Requests

- ☐ Show all requests
- ☒ Show requests pending approval
- ☐ Show approved requests
- ☐ Show completed requests
- ☐ Show rejected requests
- ☐ Show rejections in which the client has been notified

Issued Certificates

- ☐ Show all issued certificates
- ☐ Show revoked certificates
- ☐ Show suspended certificates
- ☐ Show expired certificates
- ☐ Show active certificates (not expired, not revoked, not suspended)
- ☐ Show disabled certificates (suspended or revoked, not expired)

Additional search criteria (Optional)

Requestor's name

Show recent activity only

(Not Selected) ▼

Find Certificates or Certificate
Requests

Home Page

This is the administrator's
main page.

Single Request

Requestor:	july29c	Created:	2004/07/29
Status:	Pending Approval	Modified:	2004/07/29
Transaction Id:	1jSuReRBV2VS2SHV++++++	Passphrase:	a
Template:	1-Year PKI SSL Browser Certificate	NotifyEmail:	

Previous Action Comment:

Subject: CN=wai2,OU=Class 1 Internet Certificate CA,O=The Firm
Issuer: OU=HR Cert Auth,O=IBM,C=US
Validity: 2004/07/29 00:00:00 - 2005/07/28 23:59:59
Usage: handshake(digitalSignature, keyEncipherment)
Extended Usage: clientauth

Action to take:

Action Comment (Optional)

Approve Request As It is

Approve Request with Modifications

Reject Request

Delete Request

The administrator chooses
the action on the request

Administration Home Page

Home Page

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Internet Explorer certificate install

Click "Install Certificate" to store your new certificate into your browser

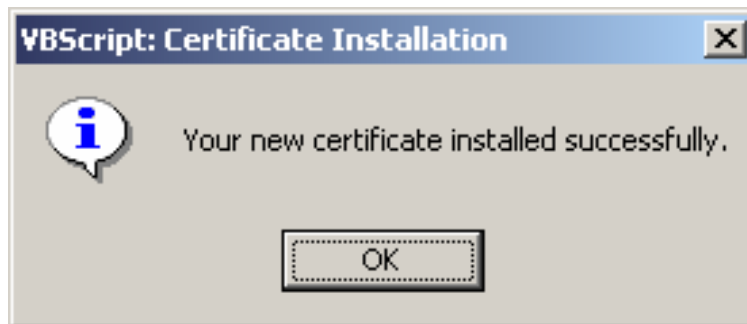
Install Certificate

Home page

After the request is approved, a certificate will be created.

If the certificate has been issued, it may be installed.

This page shows how it would look for Microsoft's IE browser



What's new in PKI Services on z/OS Release 5

- **Support Multiple Application Domains**

- To subset administration users and end users into different domains, providing different templates, functions

- **Support Certificate Suspension and Resumption**

- Certificates may be suspended for a period of time (grace period) and may be resumed later

- **New certificate extensions**

- Certificate Revocation List (CRL) Distribution Points
- Extended Key Usage
- Authority Information Access

What's new in PKI Services on z/OS Release 5...

- **Enhanced certificate extension**

- Certificate Policies

- may be created on a per template basis

- Key Usage

- create a 1-1 correspondence between key usage name and key usage bit (cf HANDSHAKE = digitalSignature + keyEncipherment)

- **Performance/Scaling Enhancements**

- Improve the performance of queries on the object store and ICL through

- Addition of VSAM Alternate Indexes
 - Setting up VSAM buffering in the PKI Services started procedure JCL
 - Use AMP= on the DD cards

What's new in PKI Services on z/OS Release 5...

- Remove certificates from Issued Certificate List (ICL) after they expired
- Replace Open Cryptographic Services (OCSF) crypto with System SSL functions
 - A transparent change

Prerequisite Products

- ▶ **RACF (or equivalent)**
- ▶ **IBM z/OS HTTP Server**
 - get working in at least non-SSL mode
- ▶ **LDAP Directory**
 - z/OS recommended - TDBM back-end required
 - Requires PKIX schema (V1R2 or higher schema)
 - Suffix must match suffix of subject DN in certificates created - This includes the CA certificate
- ▶ **Open Cryptographic Services OCSF**
- ▶ **Open Cryptographic Enhanced Plug-ins OCEP (used prior to R5)**
- ▶ **ICSF (optional)**
- ▶ **z/OS Communications Server (optional, used after R3)**

References

- **PKI Services web site:**

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

- **RACF web site:**

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

- **Cryptographic Services**

- ▶ PKI Services Guide and Reference (SA22-7693)
- ▶ OCSF Service Provider Developer's Guide and Reference (SC24-5900)
- ▶ ICSF Administrator's Guide (SA22-7521)
- ▶ System SSL Programming (SC24-5901)

- **Security Server Manuals:**

- ▶ RACF Command Language Reference (SC28-1919)
- ▶ RACF Security Administrator's Guide (SC28-1915)
- ▶ RACF Callable Services Guide (SC28-1921)
- ▶ LDAP Administration and Use (SC24-5923)
- ▶ OCEP Application Programming (SC24-5925)

- **IBM HTTP Server Manuals:**

- ▶ Planning, Installing, and Using (SC31-8690)

- **Other Sources:**

- ▶ PKIX - <http://www.ietf.org/html.charters/pkix-charter.html>

Questions???

Appendices

- How to determine if a certificate is revoked
- RACDCERT examples
- A Base64 encoded certificate request example
- PKI Services template file
- PKI Services configuration file
- PKI Services key ring
- PKI Services utilities -- vosview and iclview

How to determine if a certificate is revoked

- The application contacts the CA every time when the certificate is used. The contact information is specified in the certificate's Authority Information Access (AIA) extension.
- The CA publishes CRL to a public place, eg. LDAP server, periodically. The application checks if the certificate is on the Certificate Revocation List (CRL) published by the CA.
- As time goes, the CRL may be very large, publishing and retrieving CRL may be time consuming. Creating CRL Distribution Points to publish partial CRLs is a way to solve this problem. Again CRL Distribution Point is a certificate extension.

RACDCERT Examples

● RACDCERT GENCERT:

- ▶ Create a self signed certificate and public-private key pair using ICSF for a CA -- CERTAUTH represents the 'ID' of a CA, no 'SIGNWITH' needed for self signed cert

```
-RACDCERT CERTAUTH GENCERT SUBJECT(...) ICSF...
```

- ▶ Create a certificate and public-private key pair using PCICC for ID WEBSRV, signed with a CA certificate named 'theCA cert'

```
-RACDCERT ID(WEBSRV) GENCERT SUBJECT(...) PCICC  
SIGNWITH(CERTAUTH LABEL('theCA cert'))...
```

- ▶ Create a certificate based on a certificate request specified in the dataset 'CERTREQ.B64' for ID MYID – you get the request from other system and you want your local CA to sign it. This generates certificate only, no key pair is created.

```
-RACDCERT ID(MYID) GENCERT('CERTREQ.B64')  
SIGNWITH(CERTAUTH LABEL('theCA cert'))...
```

● RACDCERT GENREQ:

- ▶ Generate a request based on an existing certificate named 'My Self Signed Cert' in RACF for ID OUTSRV and put the request in the data set 'CERTREQ.B64'

```
-RACDCERT ID(OUTSRV) GENREQ(LABEL('My Self Signed  
Cert')) DSN(CERTREQ.B64)...
```

RACDCERT Examples...

● RACDCERT ADD:

► Add a trusted CA certificate under ID CERTAUTH

```
-RACDCERT CERTAUTH ADD('dataset containing CA cert')  
TRUST...
```

► Add a trusted peer certificate under ID SITE

```
-RACDCERT SITE ADD('dataset containing site cert')  
TRUST...
```

► Add a certificate package that contains the private key under ID myid

```
-RACDCERT ID(myid) ADD('dataset containing my cert')  
password('secret')...
```

► Replace a previous self signed certificate under ID OUTSRV

```
-RACDCERT ID(OUTSRV) ADD('dataset containing a CA  
signed cert issued for the request based on the self  
signed cert') TRUST
```

- GENCERT a self signed cert
- GENREQ based on the self signed cert
- Send request to CA and get back a new cert
- ADD the new cert back under the same ID

RACDCERT Examples...

● RACDCERT ADDRING:

- ▶ Create a key ring called 'SSLring' for ID WEBSRV

```
-RACDCERT ID(WEBSRV) ADDRING(SSLring)
```

● RACDCERT CONNECT:

- ▶ Connect WEBSRV's own certificate called 'SSL cert' to WEBSRV's key ring called SSLring

```
-RACDCERT ID(WEBSRV) CONNECT(LABEL('SSL cert')  
RING(SSLring) DEFAULT...)
```

● RACDCERT MAP:

- ▶ Create a filter to associate ID VUSER to any user presenting a certificate issued by Verisign Class 1 Individual Subscriber

```
-RACDCERT MAP ID(VUSER) IDNFILTER('OU=Verisign Class 1  
Individual Subscriber.O=Verisign, Inc.L=Internet')...
```

- ▶ Create a filter to associate ID RACFGP to any user presenting a certificate with subject's distinguished name OU=RACF.O=IBM

```
-RACDCERT MAP ID(RACFGP) SDNFILTER ('OU=RACF.O=IBM')...
```

An example of a Base64 encoded certificate request

-----BEGIN NEW CERTIFICATE REQUEST-----

```
MIIBfzCB6QIBADAQM4wDAYDVQQDEwV0ZXN0YTCBnzANBgkqhkiG9w0BAQEFAAOB
jQAwgYkCgYEA3qy4qFTb97+kefbgMysxIXpaRQVwqT0I2XeDmI0WmXF6GkvK+i4k
7wr/pto+cGtqCzHsQrm6aRKiJF6pdizYkf4xew0DqdeVArOydr/4HESVNlJRqxJ/
jqY4IJ0uphnsbKKyfl7ny77u4M50YZBXRgq9VDAFpCaQbNW8xVkiUPECAwEAAaAw
MC4GCSqGSIb3DQEJDDjEhMAAwHQYDVR0OBBYEFFX5QcyxyCL6q+NFFGjQpoCnP9mB
MA0GCSqGSIb3DQEBBQUAA4GBAMyKoRZvGJAYVPummMMiRJgMQ4KMcYrraI79rz7L
SWAq5/lPpbkue9enn1xaS3eJZOQNXGaVk4Rem3rGM740/PpQIF/qMN1pJZfOEzyL
rYxIaO6riEXM3Q2Y80m6C+X+Vk69eRC1LTvc8I5l5uz+EMCTd5x5PaGuzhXgjxkE
Q5vt
```

-----END NEW CERTIFICATE REQUEST-----

PKI Services template file, pkiserv.tmpl

Each template is divided into sections using HTML-like tags, eg.

<CONTENT>...</CONTENT>, <CONSTANT>...</CONSTANT>

- Use the CONTENT section to ask for user input, eg. to input the common name and the organization unit for the certificate

<CONTENT>

%%CommonName%%

%%OrgUnit%%

</CONTENT>

- Use the CONSTANT section to hard code predetermined values, eg. the validity period and the key usage for the certificate

<CONSTANT>

%%NotBefore=0%%

%%NotAfter=365%%

%%KeyUsage=handshake%%

</CONSTANT>

PKI Services configuration file, pkiserv.conf

Content of the configuration file includes information about:

- the name of the VSAM datasets
- whether the above datasets are shared in a sysplex
- how often a certificate and a CRL should be created
- whether email notification is requested and the information needed
- whether posting to LDAP is requested and the information needed
- whether to create the CRL Distribution Point extension and the Certificate Policies extension
- The name of the PKI Services key ring installed in RACF

PKI Services key ring

- The ring contains the CA certificate which is used to sign all the certificates generated by PKI Services
- The CA certificate can be generated by RACF or other CAs, in either case, it must be connected to a RACF key ring with usage PERSONAL and as the DEFAULT certificate
- Example:

```
RACDCERT ID(PKISRVD) CONNECT(CERTAUTH LABEL('pki ca cert')  
RING(pki_ca_ring) USAGE(PERSONAL) DEFAULT)
```

PKI Services Utilities

● vosview

- displays the records contained in the Certificate Request VSAM dataset (Objectstore)

- Sample record:

Object key = 105

name = "John Q. Public"

longkey = 1F45AEF2D3729FA35156BC47

apldata = "PKIB1YR "

comment = ""

data len = 570

flags = 1020111 - Type = Cert State = RA CertReqActive [State Flag]

PKI Services Utilities...

● iclview

- displays the records contained in the Issued Certificate List VSAM dataset (ICL)
- Sample record:

Cert 10: John Q. Public

ISSUED (Issued certificate)

Issued at 2004-05-18 14:07:05

Last changed 2004-05-18 14:07:05

Subject: CN=John Q. Public,OU=Tools Dept,O=IBM,C=US

Issuer: CN=pkica,OU=zOS Security Server,O=IBM,C=US

Requestor: John Q. Public

Appldata: "PKIB1YR "

Serial Number: 0A

Email flag: Off

Revoked at 2004-08-18 10:31:02

Revocation Reason: Temporarily suspended

End of slides