RACF and the Digital Certificate

Mark Nelson
System/390
IBM Corporation
Department BWVA Mail Station P385
Poughkeepsie, NY 12601

Federation of Security Professionals

October, 1999

Phone: (914) 435-7758
Internet: markan@us.ibm.com
IBMMAIL: USIBMV4B@IBMMAIL

Agenda

- **Review of Existing Digital Certificate Support**
  - Public key/secret key cryptography, SSL
  - Digital certificates and RACF Releases 4 and 6
  - The RACDCERT command
- **OS/390 Release 8 Security Server**
  - Extensions to the RACDCERT command
  - Key Rings
  - The "irrcerta" and "irrsitec" user IDs
  - Using Integrated Cryptographic Service Facility (ICSF) for private key storage
  - Extensions to the RACDCERT Authority Checking
- **OS/390 Release 9 Security Server**
  - Certificate Name Filtering
- **References**

Part 1 Review

Families of Cryptographic Systems

- **Symmetric or Secret Key**
  - The sender and the receiver share a secret key
  - Transmission of the key occurs "out of band" or encrypted under a different key
  - Usually more efficient that public key algorithms
- **Public Key**
  - Two keys are involved: One party uses the non-secret, told-to-everyone ***public key*** and the other party uses the private, known-only-to-its-owner ***private key***
  - The public and private keys are complementary: If one is used to encrypt data, the other is used to decrypt data
  - Public key systems can be used to verify the origination of a message as well as to verify message integrity

Message Signing

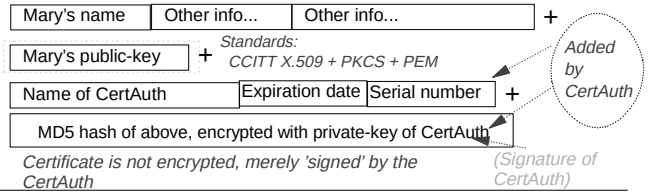
- Message origin can be determined through the use of **message signing.**
- Clear text messages are run through a **message digest algorithm** to produce a **message digest** or **hash.**
- Message digests are **one-way:** knowing the message digest reveals nothing about the message.
- If the message is sent with the message digest, then the digest can be calculated and compared at the receiving location. If the locally computed digest matches the one received with the message, then the message has not be altered. This provides message **integrity.**
- Message digests can be **signed**, that is, encrypted using a user's **private** key, providing non-repudiation.

Certificates

- **Where do I go to find someone's public key?**
  - Look in a well-known and trusted place.
    - Trusted directory
    - Public publication
  - Allow the user to tell you their public key.
    - How do you know it's really theirs?
    - How do you know that it hasn't been replaced in transit?
- **Certificate authorities** are trusted organizations which vouch for public keys.
  - Public keys are delivered via **certificates**, which are signed with the private key of the certificate authority.
  - Users manage their own certificates.

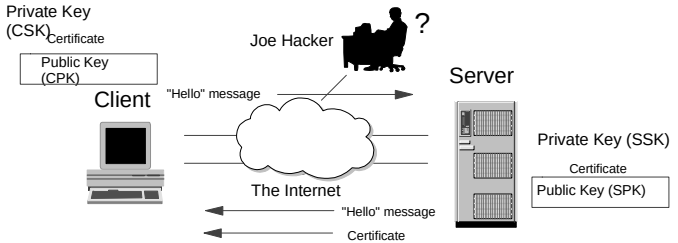
Certificates...

Mary's Digital Certificate looks about like this.



Secure Sockets Layer (SSL)...

SSL Part 1 - Kicked off by https:// URL

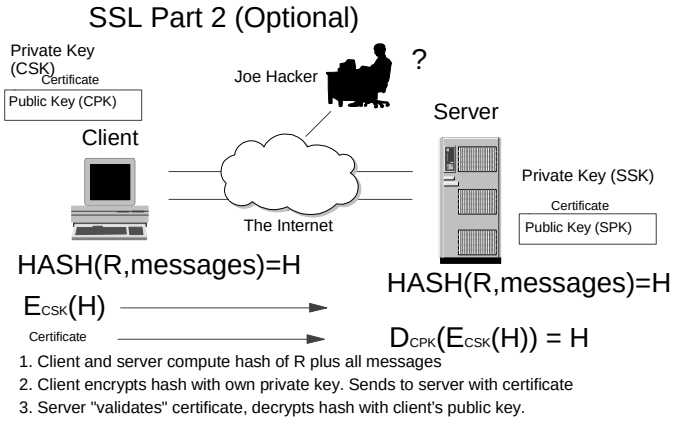


$E_{SPK}(R)$

$D_{SSK}(E_{SPK}(R)) = R$

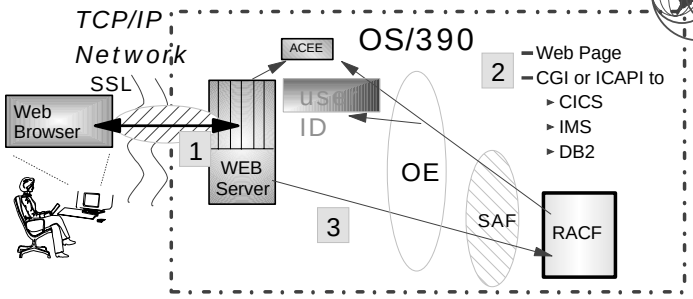
1. Client and server exchange hellos
2. Server sends certificate to client. Client "validates" certificate
3. Client encrypts random # with server's public key. Sends to server
4. Server decrypts random # with its own private key

Secure Sockets Layer (SSL)...



1. Client and server compute hash of R plus all messages
2. Client encrypts hash with own private key. Sends to server with certificate
3. Server "validates" certificate, decrypts hash with client's public key.

Digital Certificates and RACF



1- User authenticates to Secured Sockets Layer (SSL)
2- User requests OS/390 secured resource via browser
3- Web Server invokes RACF via UNIX System Services to build local security context (ACEE), passing SSL validated certificate instead of prompting for user ID & password

How does RACF Support Public Key?

- **With OS/390 Security Server R4, RACF can be used to map certificates to a RACF user ID**
  - New general resource class **DIGTCERT**
    - New segment **CERTDATA** contains the certificate
    - APPLDATA contains the user associated with the certificate
    - UACC contains the TRUST status of the certificate
    - New user profile repeat group points to the certificate
  - New RACF command **RACDCERT** to manage the certificates
- **Certificates are uniquely identified by the issuer's distinguished name and serial number**

RACDCERT Command Syntax

```
RACDCERT
   [ ID(UserID) ]
   [ LIST  [(LABEL('label-name') | [
             SERIALNUMBER(Serial-Number)
             [ ISSUERSDN('Issuer's Distinguished Name') ] ] ) ]
    | ADD('Dataset-Name')
        [ TRUST | NOTRUST ]
        [WITHLABEL('label-name')]
    | CHECKCERT('data-set-name')
    | ALTER [(LABEL('label-name') | [
             SERIALNUMBER(Serial-Number)
             [ ISSUERSDN('Issuer's Distinguished Name') ] ] ) ]
            [TRUST | NOTRUST]
    | DELETE [(LABEL('label-name') |
              [ (SERIALNUMBER(Serial-Number)
              [ ISSUERSDN('Issuer's Distinguished Name') ] ] ) ]
```

RACF
OS/390 Release 8 Security Server Support

OS/390 Release 8 Support

- **New RACDCERT functions that allow :**
  - The generation of certificates and certificate requests
  - The definition of certificate authority (CERTAUTH) and site (SITE) certificates.
  - The aggregation of certificates into key rings
  - The importation of PKCS-12 certificates
  - The renaming of the LABEL that is associated with a certificate
- **New RACF callable service to retrieve certificate information**
- **New RACF database unload (IRRDBU00) and modified RACF SMF unload (IRRADU00) records**
- **BLKUPD allows the specification of mixed case ENTRY**

GENCERT: Creating A Certificate

- **The RACDCERT GENCERT function creates a public/private key pair and a digital certificate.**
- **X.509-style keywords are used to specify certificate information, such as:**
  - Subject's distinguished name
    - Default: User's name
  - Certificate validity dates (start date/time & end date/time)
    - Default: Current date as start, one year from start date as the end date
  - Size of key
    - Range: 512-1024; Default: 512
  - Signature
    - Default: Self-signed

RACDCERT Example 1

- **RACDCERT Example #1- Create a certificate for the user ID (SRVR01) which is the user ID associated with the inventory server:**

```
RACDCERT ID(SRVR01)
         GENCERT
         SUBJECTSDN(CN('co-name.com')
                    OU('Inventory')
                     C('US'))
         WITHLABEL('Inventory Server')
```

  - Notes: SSL convention is that the "common name" (CN) is the same as the domain name

RACDCERT Example 2 - Part 1

- **RACDCERT Example #2 (Part 1)- Creating the CERTAUTH certificate**

```
RACDCERT CERTAUTH
         GENCERT
         SUBJECTSDN(CN('Local CertAuth')
                    OU('My Company')
                     C('US'))
         WITHLABEL('XYZZY CertAuth')
```

  - Note: This certificate is a certificate authority certificate (CERTAUTH), which can be used to sign other certificates.

RACDCERT Example 2 - Part 2

- **RACDCERT Example #2 (Part 2)- Create a certificate for the user ID (SRVR01) which is the user ID associated with the inventory server, this time signed by a certificate authority certificate managed by RACF:**

```
RACDCERT ID(SRVR01)
         GENCERT
         SUBJECTSDN(CN('co-name.com')
                    OU('Inventory')
                     C('US'))
         WITHLABEL('Inventory Server')
         SIGNWITH(CERTAUTH
             LABEL('XYZZY CertAuth'))
```


A Word About Distinguished Names...

- **X.509 certificates are identified by distinguished names, which are multi-part hierarchical names**
- **Distinguished names consist of these parts:**
  - Common name (CN), e.g. "Lamont Cranston"
  - Title (T), e.g. "RACF Developer"
  - One or more organizational units (OU), e.g. "RACF Development", "S390 Development", "Server Group"
  - Organization (O), e.g. "IBM Corporation"
  - Locality (L),e.g. "Poughkeepsie"
  - State or Province (SP), e.g. "New York"
  - Country (C), e.g. "US"
- **Think of the distinguished name as a hierarchical name**
  - Lamont Cranston\RACF Developer\RACF Development\S390 Development\Server Group\IBM Corporation\Poughkeepsie\New York\US

GENREQ: Creating a Certificate Request

- **RACDCERT GENREQ can be used to create a certificate request which can be submitted to an off-platform certificate authority to sign the certificate.**

```
RACDCERT ID(SRVR01)
         GENREQ(LABEL('Inventory Server'))
         DSN('MARKN.INVSERV.GENREQ')
```

EXPORT: Writing a Certificate to a Data Set

- **RACDCERT EXPORT can be used to extract a certificate from the RACF data base and place it into a data set**

```
RACDCERT ID(SRVR01)
         EXPORT(LABEL('Inventory Server'))
         DSN('MARKN.INVSERV.GENREQ')
```

- **One in a data set, the certificate can be moved to the OS/390 HFS, where it can be loaded by a web browser for use on a client system.**

Key Rings

- **Certificates are collected into sets called *key rings.***
  Key facts about key rings:
  - Each key ring is associated with an OS/390 user ID.
  - User IDs may have more than key ring.
  - CertAuth and Site certificates are associated with the "reserved" user IDs "irrcerta" and "irrsitec"
  - Key rings are identified by a 1 to 237 character ring name
  - Authority to manage key rings may be distributed. However, the system security administrator has the ability to define the superset of key ring contents.

Advantages of RACF Key Rings

- **Why are RACF key rings superior to other platforms key ring implementations?**
  - Application/server owners may implement a trust hierarchy that is a subset of the installation policy. That is, they may not allow non-approved certificate authorities.

ADDRING: Creating a Key Ring

- **RACDCERT ADDRING is used to create a key ring. To create a key ring called "RING01" for the user ID SRVR01, the command is:**

```
RACDCERT ID(SRVR01)
         ADDRING(RING01)
```

CONNECT: Adding a Certificate to a Key Ring

- RACDCERT CONNECT is used to add a certificate to a key ring. To connect the certificate labeled "Inventory Server" to a key ring called RING01 that is associated with the user ID SRVR01, the command is:

```
RACDCERT ID(SRVR01)
         CONNECT(
             LABEL('Inventory Server')
             RING(RING01)
             DEFAULT)
```


CONNECT: Example 2

- To connect a certificate authority certificate to a key ring, the CERTAUTH keyword is added to the RACDCERT CONNECT command:

```
RACDCERT ID(SRVR01)
         CONNECT(CERTAUTH
             LABEL('CertAuth 1')
             RING(RING01)
             DEFAULT)
```


REMOVE: Taking a Cert out of a Key Ring

- RACDCERT REMOVE is used to take a certificate out of a key ring. To remote the certificate labeled "Inventory Server" from key ring RING01 for the user ID SRVR01, the command is:

```
RACDCERT ID(SRVR01)
         REMOVE(
             LABEL('Inventory Server')
             RING(RING01))
```


DELRING: Deleting a Key Ring

- RACDCERT DELRING is used to delete a key ring. To delete the key ring called "RING01" for the user ID SRVR01, the command is:

```
RACDCERT ID(SRVR01)
         DELRING(RING01)
```

- DELRING does not delete the certificates themselves. It merely deletes the relationship between the certificate and the key ring.

LISTRING: Listing a Key Ring

- **RACDCERT LISTRING is used to lis the contents of a key ring. If a listing of all rings associated with a user ID is desired, then LISTRING(*) is specified.**
- **LISTRING displays:**
  - The ring name
  - The label of the certificate
  - The DEFAULT status of the certificate within the ring
  - The usage within the ring

LISTRING: Sample Output

```
Digital ring information for user GEORGEM:

  Ring:
       >GEORGEMsNewRing01<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  --------------------------------   ------------   --------   -------
  New Cert Type - Ser # 00            ID(GEORGEM)    PERSONAL     YES
  New Type Cert - VsignC1             ID(GEORGEM)    CERTAUTH     NO
  New Type Cert - VsignC2             ID(GEORGEM)    SITE         NO
  65                                  ID(JOHNP)      PERSONAL     NO

  Ring:
       >GEORGEMsRing<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  --------------------------------   ------------   --------   -------
  GEORGEM's Cert # 48                 ID(GEORGEM)    PERSONAL     NO
  GEORGEM's Cert # 84                 ID(GEORGEM)    PERSONAL     NO
  New Cert Type - Ser # 00            ID(GEORGEM)    PERSONAL     YES

  Ring:
       >GEORGEMsRing#2<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  --------------------------------   ------------   --------   -------
  GEORGEM's Cert # 84                 ID(GEORGEM)    PERSONAL     NO
  GEORGEM's Cert # 48                 ID(GEORGEM)    PERSONAL     NO

  Ring:
       >GEORGEMsRing#3<
*** No certificates connected ***
```


A Word About "irrcerta" and "irrsitec"

- **The two special user IDs "irrcerta" and "irrsitec" are anchor points for certificate authority certificates and site certificates respectively. Key points about these IDs:**
  - They are marked as REVOKED in the RACF data base
  - RACROUTE REQUEST=VERIFY requests fail for these user IDs as they have no default group
  - ADDUSER, DELUSER, and LISTUSER may not be used against these specific IDs. Note that a "LISTUSER *" will list the information about these IDs
  - The SEARCH command for CLASS(USER) will return these user IDs if they fall within the SEARCH criteria
  - ICHEINTY NEXT and RACROUTE EXTRACTN processing will return these IDs if they match the selection criteria

Remote Sharing Considerations

- **RRSF does not propagate private key information**
  - CERTPRVK (private key)
  - CERTPRVS (private key size)
  - CERTPRVT (private key type)
- **Customers must ensure that RRSF has identical propagation rules for the DIGTCERT and DIGTRING class**
  - Recommendation: Define the propagation with the profile AUTODIRECT.node.DIGT*.APPL to have a single set of rules for both classes.

ICSF Considerations

- **IBM recommends the use of the S/390 Integrated Cryptographic Support Facility (ICSF) for the storage of private keys.**
- **ICSF ensures that the user's private key is stored within ICSF, encrypted under the ICSF master key for the installation**
- **If the RACF database is shared among systems, then all of the ICSFs must have the same master key**
  - Master keys may be managed using the Trusted Key Entry (TKE) workstation
- **ICSF is not required; it is used if available (and configured)**
  - If ICSF is not being used, BSafe (software encryption) is used

ICSF Considerations...

- **ICSF-stored private keys are requested by using the "ICSF" keyword on RACDCERT GENCERT and RACDCERT ADD.**
- **Non-ICSF-managed private keys may be moved into ICSF storage by:**
  - RACDCERT EXPORTing the certificate to a data set and then
  - Re-ADDing the certificate specifying the "ICSF" keyword.
    - Since the subject's distinguished name, public key, and issuer's distinguished name are the same, RACF replaces the certificate, and migrates the private key to ICSF
    - Note that the reverse process is not possible.

Authority Checking

- **Users with SPECIAL authority can do practically anything to anybody.**
- **For everyone else, authority checks are performed against the resource IRR.DIGTCERT.<function>. The authority required to this resource is:**
  - READ to perform the function on their own certificate or key ring,
  - UPDATE to perform the function on the certificate or key ring of another, and
  - CONTROL to perform the function on a certificate authority or site certificate.

Authority Checking Summary

Function	READ	UPDATE	**CONTROL**
ADD	Add a cert to one's own user ID	Add a cert to someone else's ID	**Add a SITE or CERTAUTH cert**
ALTER	Change the trust status or label of one's own cert	Change the trust status or label of someone else's cert	**Change the trust status or label of a SITE or CERTAUTH cert**
DELETE	Delete one's own cert	Delete someone else's cert	**Delete a SITE or CERTAUTH cert**
EXPORT	**Export one's own cert**	**Export someone else's cert**	**Export a SITE or CERTAUTH cert**
GENREQ	**Generate a request based on one's own cert**	**Generate a request based on someone else's cert**	**Generate a request based on a SITE or CERTAUTH cert**
LIST	List one's own cert	List the someone else's cert	**List a SITE or CERTAUTH cert**

BOLD indicates new with Release 8

Authority Checking Summary...

Function	READ	UPDATE	CONTROL
ADDRING	Create a key ring for one's own ID	Create a key ring for someone else's ID	N/A
CONNECT	Place ones own certificate in one's own ring	Place a CERTAUTH or SITE cert in one's own ring	Place a certificate into someone else's ring
DELRING	Delete one's own key ring	Delete someone else's key ring	N/A
LISTRING	List one's own key ring	List someone else's key ring	N/A
REMOVE	Remove a certificate from one's own key ring	Remove a SITE or CERTAUTH certificate from one's own key ring	Delete a certificate from the ring of another

Changes to IRRDBU00 Output

- **Changes to database unload utility (IRRDBU00)**
  - Record type 0560 ("Certificate Data Record") is now unloaded. This record contains:
    - Start and end dates and times for the certificate
    - The type of private key associated with the certificate. Valid values are PKCSDER, ICSFTOKN, NONE, and UNKNOWN
    - The size of the private key, expressed in bits
    - The hexadecimal representation of the 8 byte serial number of the last certificate signed with this key
    - The ring sequence number
  - RACDBUTB and RACDBULD have been updated in 'SYS1.SAMPLIB'

Changes to SMF Records

- **Changes RACF's Type 80 Record:**
  - For event code 66, relocate section 6 contains the new RACDCERT keywords and values
  - Eight new relocate sections have been created:
    - 320: Ring name
    - 321: SUBJECTSDN country value ("C")
    - 322: SUBJECTSDN state or province value ("SP")
    - 323: SUBJECTSDN locality value ("L")
    - 324: SUBJECTSDN organization value ("O")
    - 325: SUBJECTSDN organizational unit value ("OU")
    - 326: SUBJECTSDN title value ("T")
    - 327: SUBJECTSDN common name ("CN")

Changes to IRRADU00 Output

- **IRRADU00 unloads the new RACDCERT keywords**
- **No new IRRADU00 record types**
- **No existing IRRADU00 record formats are altered**

New Callable Service: IRRSDL00

- **RACF is providing a new callable service, IRRSDL00, for use in implementing the Common Data Security Architecture (CDSA) Data Library (DL) functions**
- **IRRSDL00 is a key-8, non-APF, problem state programming interface**
- **Access to IRRSDL00 functions are controlled by checks against the IRR.DIGTCERT.<function> resources in the FACILITY class**
  - READ to IRR.DIGTCERT.LISTRING to retrieve one's own
  - UPDATE to IRR.DIGTCERT.LISTRING to retrieve someone else's
- **The private key or private key label that is associated with the certificate may not be retrieved unless the execution user ID is equal to the user ID that is associated with the certificate.**

UNIX Programming Interface

- **Open Cryptographic Services Facility**
  - **Keyworks (CDSA) on OS/390**
  - **Provides a framework for a base set of cryptographic functions**
    - **Encrypt, decrypt, generate keys, etc**
  - **Service Provider Modules ("Plug-ins") perform the functions**
- **Open Cryptographic Enhanced Plug-ins**
  - **Datalib and Trust Policy Plug-ins that make use of certificates/ keys stored in RACF**

Exploiters

- **The following products make use of RACF's certificate support:**
  - Client support only:
    - IBM HTTP Server
    - CICS
  - Client support and server support through System SSL:
    - LDAP
    - Host on Demand
  - Server support using OCEP:
    - Firewall

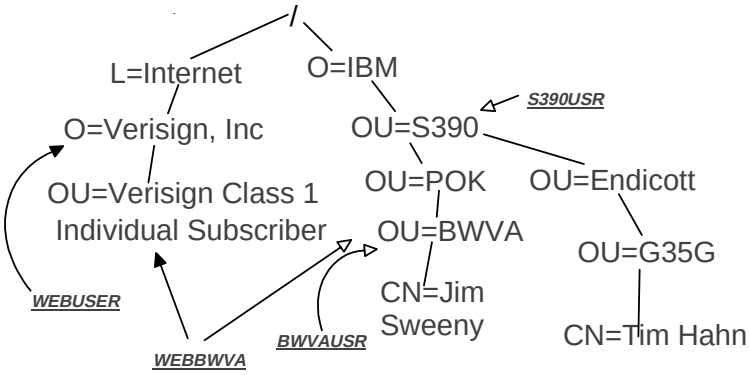
RACF
OS/390 Release 9 SecureWay Security Server Support

Certificate Name Filtering

- **Usability and Functional Enhancement to RACF Digital Certificate Support**
  - Certificate used to have to be installed in RACF for each user, can now use profiles based on issuer's and subject's distinguished names to choose the user ID to assign for a particular certificate
    - ►Greater accountability/audit capabilities
    - ►Greater granularity of access control
- **OW40129 - Shipped in R9 timeframe on R8 and R9 along with accompanying SAF APAR**

Examples

X.500 Directory Information Tree (DIT) , example:



X.500 Distinguished Names. . .

- **Hypothetical certificates containing:**
  - **Subject Name** - **CN=Jim Sweeny.OU=BWVA.OU=POK.OU=S390.O=IBM**
  - **Issuer Name** - **OU=Verisign Class 1 Individual Subscriber.O=Verisign, Inc.L=Internet**
  - **Subject Name - CN=Tim Hahn.OU=G35G.OU=Endicott.OU=S390.O=IBM**
  - **Issuer Name - OU=Verisign Class 1 Individual Subscriber.O=Verisign, Inc.L=Internet**

Solution (Part 1)

- **Subject and Issuer Name Filtering**
  - Create DIGTNMAP class and profiles to...
    - ►Assign User ID based on Subject's placement in DIT
    - ►Assign User ID based on Issuer's Name
    - ►Assign User ID based on both the Subject's placement in DIT and full Issuer's Name
  - Profile form: subject-filter<delimiter>issuer-filter
    - ►New field FLTRUSER contains User ID

InitACEE Processing

- **Do old style certificate lookup first**
  - If not found search for Map profiles more specific to least specific:
    1. Iteratively-shrinking-subject-name with full-issuer-name
    2. Iteratively-shrinking-subject-name alone
    3. Iteratively-shrinking-issuer-name alone
- **Get User ID from FLTRUSER of first match**

Solution (Part B)

- **Additional Criteria For User ID Determination**
  - Allow a given name filter to map to multiple User IDs
  - Actual User ID determined by system and/or user variables (APPLID, SYSID, and new InitACEE variable list parameter)
    - ►FLTRUSER contains profile template containing variable names
    - ►Variable values substituted to form real profile name
    - ►Get UserID from APPLDATA field of new DIGTCRIT Class profile

RACDCERT Command Changes

```
RACDCERT [ID(user-id) | MULTIID]
  MAP [('cert-dsn')]
   [SDNFILTER('subject-dist-name-filter')]
   [IDNFILTER('issuer-dist-name-filter')]
   [CRITERIA('criteria-profile-name-template')]
   [WITHLABEL('label-name')]) [TRUST | NOTRUST]

  LISTMAP (LABEL('label-name'))

  ALTMAP (LABEL('label-name'))
   [NEWCRITERIA('criteria-profile-name-template')]
   [NEWLABEL('label-name')] [TRUST | NOTRUST]

  DELMAP (LABEL('label-name'))
```


Access Control / Auditing

- **InitACEE CREATE would save full Subject's and Issuer's name off ACEE**
  - New RACROUTE VERIFY parameter, X500NAME
- **InitACEE QUERY may return X500NAME in addition to User ID**
- **RACHECK Exit can retrieve X500NAME**
- **Subject's and Issuer's names recorded on every audit record (max 255 bytes each)**

Reference Material

Reference Material

Changes to RACDCERT for R8

- **Changes to RACDCERT ADD**
  - SITE and CERTAUTH certificates may now be added.
  - Replacement certificates may be added. This allows
    - Replacement certificates to be added
    - Certificates returned from certificate requests:
  - PKCS#12 "certificate packages" (which contain a private key) may be RACDCERT ADDed

Changes to RACDCERT for R8...

- **Changes to RACDCERT CHECKCERT**
  - SITE and CERTAUTH certificates may now be CHECKCERTed
  - CHECKCERT now reports if a certificate is defined as a certificate authority certificate or a site certificate; This is done only after checking IRR.DIGTCERT.LIST in the FACILITY class
  - CHECKCERT may now be used to check a PKCS#12 certificate package

Changes to RACDCERT for R8...

- **Changes to RACDCERT ALTER:**
  - SITE and CERTAUTH certificates may now be ALTERed
  - ALTER now supports the altering of the label of a certificate through the use of the LABEL and NEWLABEL keyword
- **Changes to RACDCERT DELETE:**
  - SITE and CERTAUTH certificates may now be DELETEd
- **Changes to RACDCERT LIST:**
  - SITE and CERTAUTH certificates may now be LISTed
  - LIST now displays information about the private key (type and size

LIST: Sample Output

```
Digital certificate information for user GEORGEM:

  Label: New Cert Type - Ser # 00
  Status: TRUST
  Start Date: 1996/04/18 03:01:13
  End Date:   1998/02/13 03:01:13
  Serial Number:
       >00<
  Issuer's Name:
       >OU=Internet Demo CA.O=Xcert Software Inc.<
  Subject's Name:
       >OU=Internet Demo CA.O=Xcert Software Inc.<
  Private Key Type: ICSF
  Private Key Size: 1024

Ring Associations:
  Ring Owner: GEORGEM
  Ring:
     >GEORGEMsNewRing01<
  Ring Owner: GEORGEM
  Ring:
     >GEORGEMsRing<
```


References

- ***Network Security, Private Communication in a Public World,*** Charlie Kaufman, Raida Perlman, and Mike Speciner, Prentice Hall, 1995
- ***Lotus Domino Go Webserver/Internet Connection Secure Server/Websphere Application Server Webmaster's Guide,*** IBM Corporation, various dates
- ***The Codebreakers: The Story of Secret Writing,*** David Kahn, Macmillan, 1997
- ***RSA Developer Support Web Site,*** RSA Data Security Inc., http://support.rsa.com/
- ***Cryptography Frequently Asked Questions (FAQ)***, Internet

References...

- **RACF Command Language Reference (SC28-1919)**
- **RACF Macros and Interfaces (SC28-1914)**
- **RACF Security Administrator's Guide (SC28-1915)**
- **RACF Auditor's Guide (SC28-1916)**
- **RACF Callable Services Guide (SC28-1921)**
- **OS/390 Security Server Open Cryptographic Enhanced Plug-ins (OCEP) Guide and Reference (SA22-7429)**
- **OS/390 OCEP Module Developer's Guide and Reference (SC24-5876)**
- **OS/390 OCEP Application Developer's Guide and Reference (SC24-5875)**

References...

- **RACF Web Page: http://www.ibm.com/s390/racf**
  - **Latest release information on RACF**
  - **Links to announcement letters**
  - **Sample code**
    - **DBSYNC to compare two RACF data bases**
    - **RACFDB2 to migrate DB2 access control to RACF**
    - **RACFICE to create reports**
    - **OS390ART for a web-based reporting tool**
    - **RACTRACE tracing facility**
  - **Frequently asked questions (FAQ)**
  - **RACF user group information**

Summary

- **Digital certificates are the backbone of the security for e-business arena applications**
- **RACF's support for server-side digital certificates is a natural evolution of the existing RACDCERT support, providing:**
  - ►Confidentiality of private keys
  - ►An implementation of key rings that allows central control of the security policy
  - ►Callable service support which enables CDSA-support (OCSF) for the RACF-managed certificate data