IBM Systems

Getting started with Digital Certificates Part I

This is a two part presentation where we will attempt to unlock the mysteries of digital certificates. Part I will get you started in the world of Digital Certificates. We will explain what they are, the 'keys' involved in digital certificates, what is their importance, the information contained in a digital certificate and an example

Legal Disclaimer

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This information could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) at any time without notice.

Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

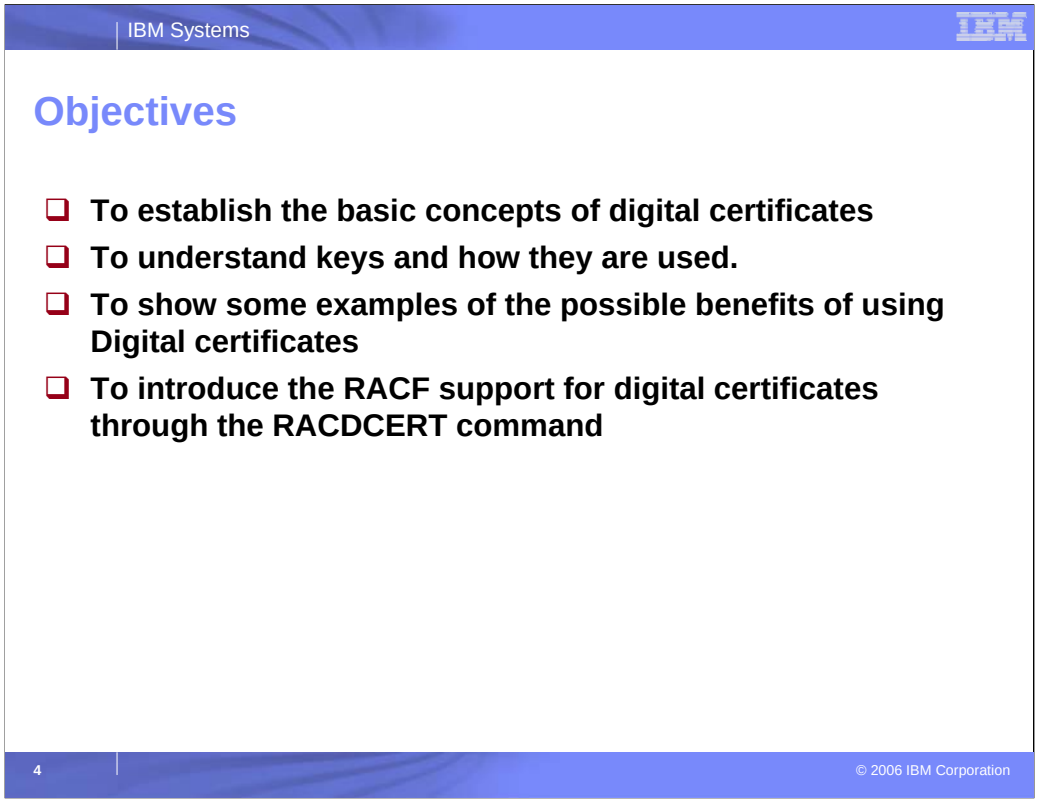
The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

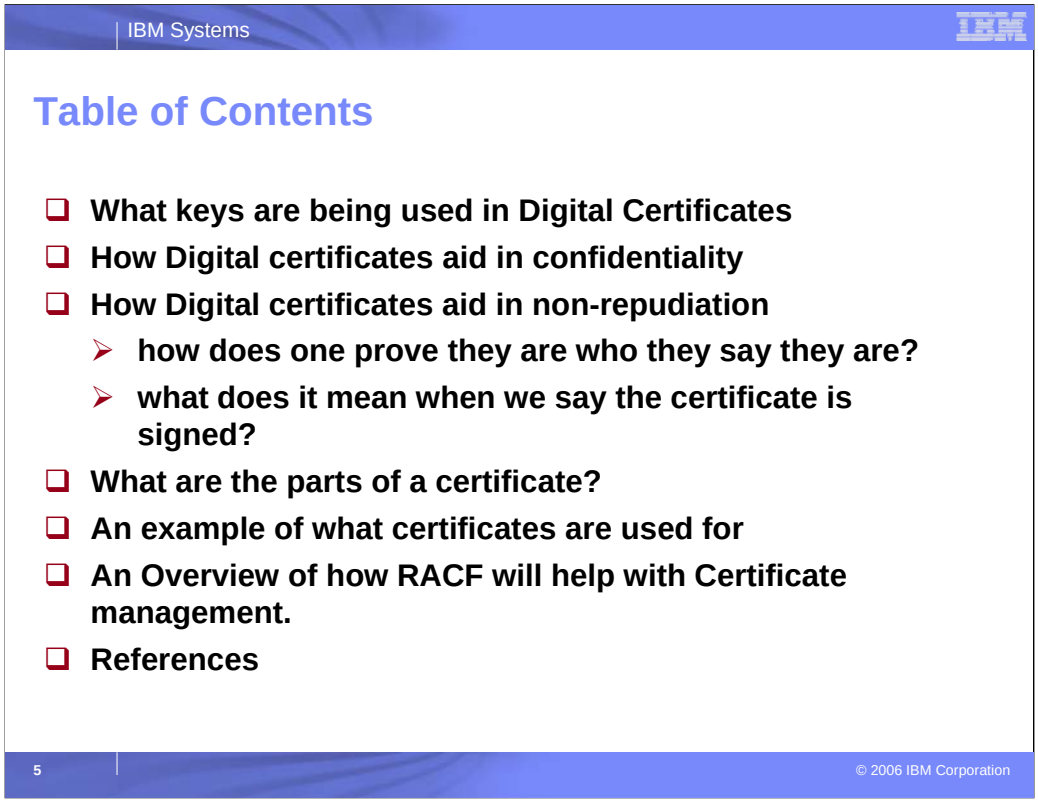
Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

For a list of trademarks, go to this url.



- Digital certificates will allow you to have a unique method of identifying the user. Each certificate will have certain information regarding the user plus be verified by an authority that who the user says they are, can be believed.
- We will look a the mechanics of how Digital certificates will allow users the freedom of confidentiality, integrity and non-repudiation.
- By showing a practical example, you will understand these mechanics.
- We will also discuss in a high level how the RACDCERT command in RACF will help you learn how to create, use and manage Digital Certificates.



•What is Public Key Cryptography. How does it work? We will discuss this and how it is used. We will start with what we mean by keys and what types of keys we will use.

•We will take a look at how digital certificates will allow you to send a message to a recipient and to know that the recipient only is the one who can read this message.

•We will discuss non-repudiation and how the recipient can be assured that the message received came from the correct sender and how we can be assured that the message has not been altered or tampered with.

•We will take a look at some of the parts of a digital certificate.

•We will follow a practical example of how this all works together.

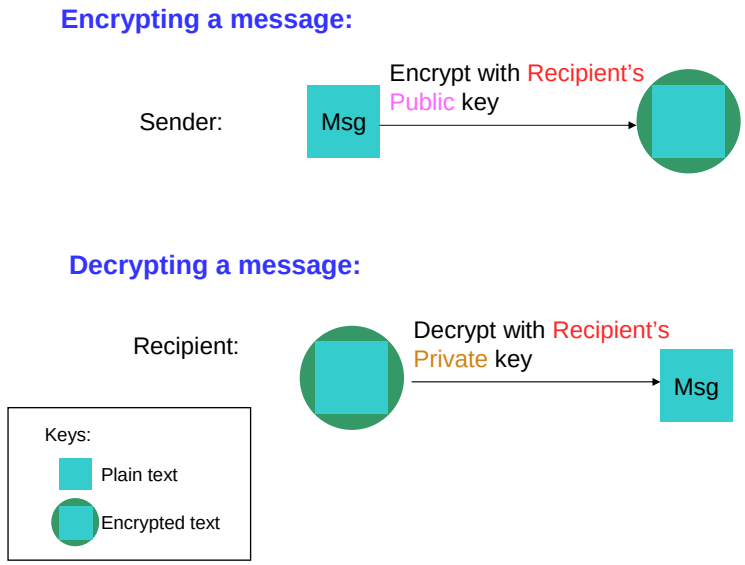
•And, finally, we will look at an overview of how RACF can work in the creation and management of digital certificates.

Public Key Cryptography overview

- **Secret Key (Symmetric Key) cryptography encrypts and decrypts with the same key**
- **Public Key (Asymmetric Key) cryptography involves a public-private key pair, encrypts with one key and decrypts with its partner key**
- **Public Key cryptography can be used to facilitate goals of communication in a security-rich environment**
  - **Confidentiality**
  - **Integrity and Non-repudiation**

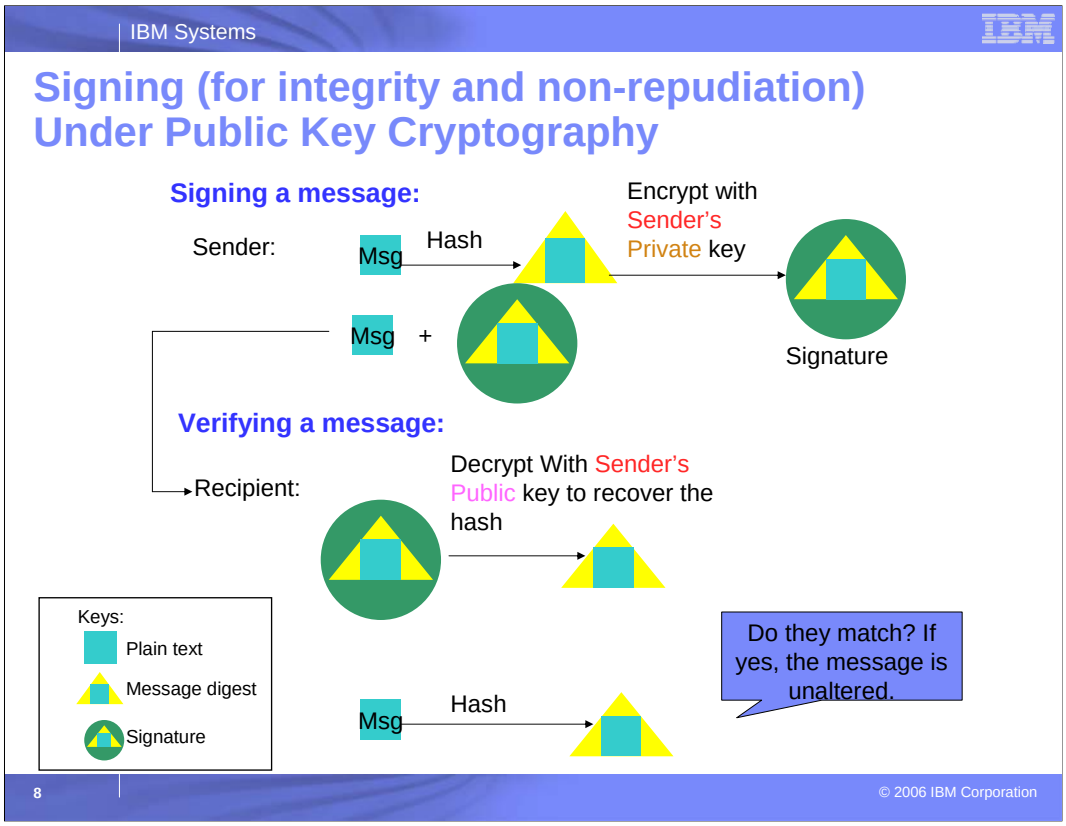
•Secret keys, also known as symmetric keys, is one key that a user would use to encrypt and decrypt. However, in order for someone else to decrypt the message, they must also have the secret key. This causes some problems, how do we get the secret key to the recipient?, can we be confident that the key has not been compromised (given to someone else)?, how do we manage renewals?, etc..
•Let's discuss another type of key, public-private key pairs, also known as asymmetric keys . Public/Private key cryptography involves a function in which encryption and decryption are using associated keys from a mathematically related key pair. Something encrypted with one key (let's say private key) can only be decrypted by the other key (public key).
•Public/private keys are the mainstay of a digital certificate.
•Let's go through a demonstration of public/private keys.

Encryption (for confidentiality) Under Public Key Cryptography



1. The sender wants to send a confidential message to a particular recipient. [enter]
2. To do so, the sender will encrypt the message with the recipient's public key. [enter] During encryption, a certain encryption algorithm is used. The sender and the recipient have to communicate the algorithm that they are using so that the recipient can use the same algorithm to decrypt. This algorithm sharing can be done with the use of asymmetric keys; public and private key pairs. To send a confidential message, the sender will encrypt their own message with the recipient's public key.[enter]
3. When the recipient receives the message, [enter] they will decrypt with their corresponding private key to see the message. The recipient's private key is known only to them.

This method ensures confidentiality from the sender to the recipient.



Let's now look at a way that the sender can verify that the message they are sending cannot be tampered with.

[enter]

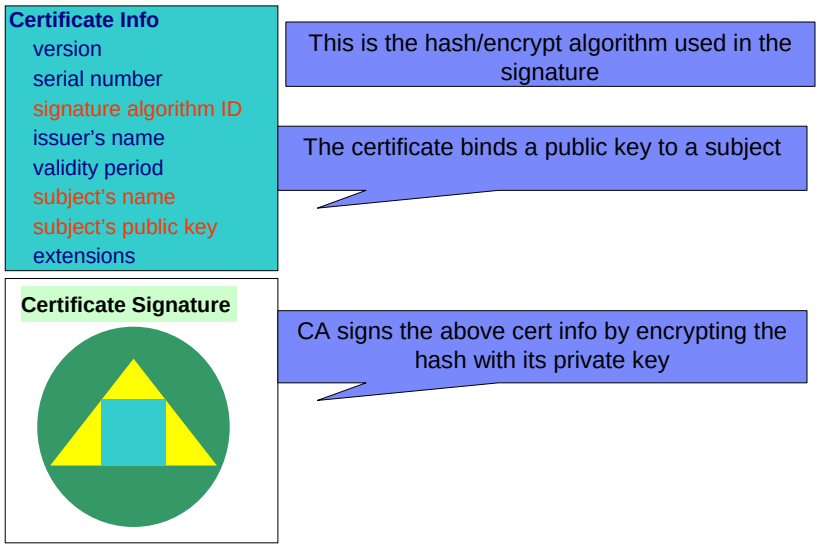
- The sender wants to send a message and ensure that it has not been altered. Again, this can be done with the use of asymmetric keys. [enter]
- The sender will hash the message. [enter]
- They will then encrypt the hash with their own private key. The signature is produced after the hashing and encryption processes. This also requires the sender to communicate the public key to the recipient so that he can verify the signature. [enter]
- We call the combined algorithms the signing algorithm. [enter]
- To verify the message, the recipient receives the message plus the signed hash. [enter]
- The recipient will then decrypt the signature with the sender's public key to recover the hash. [enter]
- The matching of the hashes not only proves that the message has not been altered, it also proves that the sender signed the message and the sender cannot deny it.

Public Key and Digital Certificate

- **Public Key by itself does not identify its owner**
- **Need a trusted third party, known as Certificate Authority (CA), to bind the Public Key to a subject through a Digital Certificate**
- **The Certificate Authority signs the Digital Certificate with its Private Key to prove its authenticity**

•Each certificate will have more information than just the public key. There is additional information about the owner (subject). We will look at this information shortly.
•There also needs to be an authority that will vouch that the information about the subject is correct. The certificates issued by the popular Certificate Authorities (CA) are widely accepted since those CA certificates were installed in most of the applications that use certificates, like the browser, ftp programs…
•Each certificate is unique. The Certificate Authority will sign each certificate and be known as the Issuer of that certificate. Each certificate will have the combination of Serial Number and Issuer's name. That combination is unique. For example, there will only be one instance of a certificate signed by Acme Certificate Authority with the serial number of 123456.

What's inside a Certificate?



The contents of a certificate may be information such as the owner of the certificate (Subject's Distinguished Name); the Serial Number; the signer of the certificate (Issuer's Distinguished Name); the validity period (the starting and ending dates when the certificate is considered valid); the hashing algorithm used; and extensions. There are many types of extensions – Key usage, Subject Alternate Name, Certificate Revocation List (CRL) Distribution Points…Different applications may require different extensions.

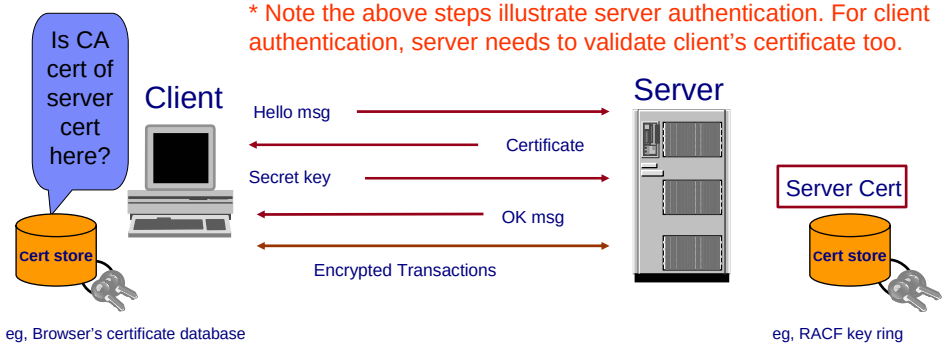
•Subject's name and issuer's name are in the x500 format, like CN=Wai Choi.OU=RACF.O=IBM.C=US

•Examples of signature algorithm: RSA with MD5, RSA with SHA1, DSA with SHA1.

A common use of Certificate – 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.



•Now let's look at a practical example. The client's browser and the host site negotiate what is called a secure Socket Layer (SSL) handshake. [enter]

•Your browser (Client) will notify the server that it wants to communicate. [enter]

•The server will send it's own certificate to your browser in response to prove its identity and ensure you that you are dealing with the correct server. [enter]

•Your browser then determines whether it can trust that certificate. It validates this by checking whether a certificate for the issuer is in your browser's trusted certificate store or key ring. Both the server and the client need to have a certificate store to hold the certificates. The certificate store is also referred to as a key ring. [enter]

•If so, a secret key is generated to encrypt the traffic between the client and the server because encryption/decryption is more efficient with symmetric key. The client uses the server's public key from the server's certificate to encrypt its own secret key and sends it back to the server. The public key is used to help the safe transportation of the secret key to the server. [enter]

•The server can then decrypt the client's secret key with it's own private key. [enter]

•The server will then encrypt a message back to the client to let the client know that it is OK to start. [enter]

•Client trusts server, business can be conducted. [enter]

•The above steps illustrate server authentication. For client authentication, server needs to validate the client's certificate also.

Helpful Terms

- Digital Certificate
- IssuersDN
- SubjectsDN
- Asymmetric Keys
- Symmetric Keys
- Key Ring
- BER encoding
- DER encoding

Here are some helpful terms to review:

Digital Certificate – A data structure that represents the binding of a user's distinguished name and a public key. Certificates are typically signed to enable a recipient of the certificate to ensure that the binding of the user's distinguished name (DN) and public key have not been tampered with.

Issuer's Distinguished Name – or IDN. The X.509 name that is associated with a certificate authority.

Subject's Distinguished Name – or SDN The X.509 name in a digital certificate that is associated with the name of the subject.

Asymmetric Key – also known as Public key cryptography involves a public-private key pair, encrypts with one key and decrypts with its partner key.

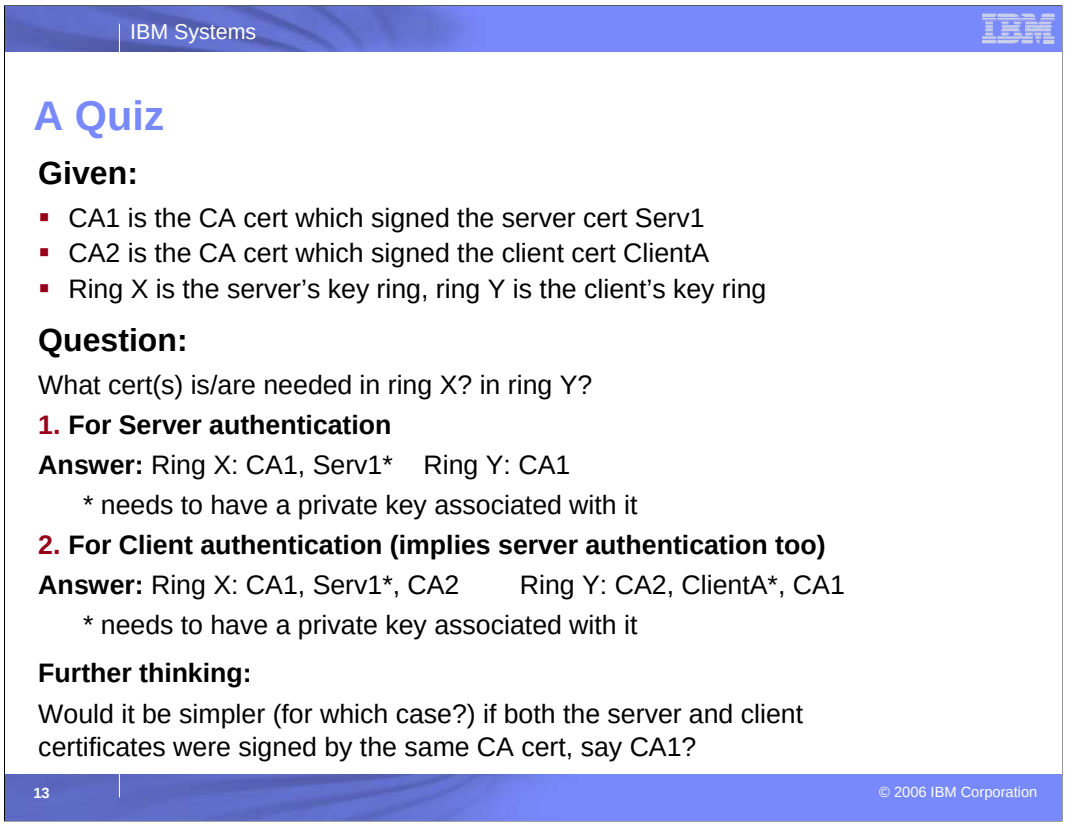
Symmetric Key – also known as Secret Key cryptography encrypts and decrypts with the same key.

Key Ring – a named collection of personal and certificate-authority certificates for a specific user. Each key ring represents a different trust policy in effect for the user.

Some other terms used in discussing certificates are:

BER – This term represents the Basic Encoding Rules specified in ISO 8825 for encoding data units described in abstract syntax notation 1 (ASN.1). The rules specify the encoding technique.

DER – This term represents the Distinguished Encoding rules, which are a subset of the Basic Encoding Rules (DER).



•Both the server and the client need to have a key ring, even in the case of server authentication only.

•If the certificate is used to prove its identity to the other party, it needs a private key associated with it.

•If the certificate is used to verify a certificate coming from the other party, it doesn't need a private key associated with it.

•Looking at the certificates and the rings we have for the server and the client, what certificates are needed in Ring X and in ring Y for Server authentication? [enter]

•For Server authentication, the Server's key ring, ring X needs to have Serv1, the server's certificate with a private key and CA1, the Certificate Authority that signed Serv1 . Ring Y, the client's ring would only need to have CA1, the Certificate Authority that signed Serv1. If a client wants to do server authentication, he just needs to get the CA certificates that he trusts in his key ring. [enter]

•For question 2, Client Authentication; [enter]

•In order to do client authentication, the client needs to get his own certificate from a CA and place the certificate in his (client) key ring. Usually the certificate of the CA which issued the client's certificate will be placed in the key ring also.

•So, the Server's key ring, ring X will have the Server's certificate with private key, Serv1; CA1 who signed Serv1; and the CA that signed the client's certificate, CA2.

•The Client's ring, Ring Y will have his own certificate (ClientA) with a private key; possibly the CA that signed his certificate, CA2, and the Certificate authority

Now we will take a high level look at the way that RACF on z/OS can help you with digital certificate management.

Basic RACDCERT functions

- **Certificate Management:**
  - RACDCERT GENCERT – generate and install a certificate
  - RACDCERT GENREQ – generate a certificate request
  - RACDCERT ADD – install a certificate
  - RACDCERT LIST – display certificate information from an installed certificate
  - RACDCERT ALTER – change certificate installation information
  - RACDCERT DELETE – erase a certificate
  - RACDCERT CHECKCERT – display certificate information from a dataset
  - RACDCERT EXPORT – export a certificate
  - RACDCERT REKEY – renew certificate with new key pair
  - RACDCERT ROLLOVER – finalize the REKEY process

For certificate management, we will need to have the ability to create certificates, certificate requests (a request that will be sent and signed by a Certificate Authority), and add already signed certificates into the RACF database. We do this with the RACDCERT command and the functions of GENCERT, GENREQ and ADD, respectively. To remove a certificate for a user from the RACF database, use the function of DELETE. To look at what certificates are installed for a specific user in the RACF database, you could use either the LIST or CHECKCERT functions. To change the status or the label of an installed certificate, use the ALTER function. To write a certificate to a z/OS dataset, use the EXPORT function. If the current certificate needs to be renewed and a new key pair created, use the REKEY function followed by the ROLLOVER function to finalize the REKEY process.

Advanced RACDCERT functions

- **Certificate installation (Rings)**
  - RACDCERT ADDRING – create a key ring
  - RACDCERT CONNECT – place a certificate in a key ring
  - RACDCERT REMOVE – remove a certificate from a key ring
  - RACDCERT LISTRING – display key ring information
  - RACDCERT DELRING – delete a key ring
- **Certificate Mapping**
  - RACDCERT MAP – create a certificate filter
  - RACDCERT ALTMAP – change the certificate filter
  - RACDCERT DELMAP – delete a certificate filter
  - RACDCERT LISTMAP – display certificate filter information

In order the have key rings as mentioned earlier, we would use the RACDCERT command with the functions of ADDRING to add a ring to the RACF database; CONNECT to place a certificate in the key ring; REMOVE to take out the certificate in a key ring and DELRING – to delete the entire ring from the RACF database. To look at what rings and what certificates are connected to the rings for a specific user, use the LISTRING function of the RACDCERT command.

If there is a need for many certificates to map to one user or a group of different users depending on a certain level of the Distinguished Name in the certificate, certificate mapping is available through the RACDCERT command also. To create a mapping filter, use the MAP function. To change an existing filter, use ALTMAP. DELMAP will delete a certificate filter and LISTMAP will display the filter information.

Basic Rules of RACDCERT

- **Syntax: RACDCERT <ID type> <Function> <Function specific keywords>**

Entity	RACDCERT function	ID Type
Certificate	GENCERT GENREQ ADD LIST ALTER DELETE CHECKCERT EXPORT REKEY ROLLOVER	Ordinary MVS ID – ID(xxx) Certificate Authority ID - CERTAUTH External system ID - SITE
Key Ring	ADDRING LISTRING DELRING	Ordinary MVS ID – ID(xxx)
Key Ring and Certificate	CONNECT REMOVE	Ordinary MVS ID – ID(xxx)
Certificate Filter	MAP LISTMAP ALTMAP DELMAP	Ordinary MVS ID – ID(xxx) Multiple mapping ID - MULTIID

•This presentation just gives you a high level of the RACDCERT command. This chart indicates the overall syntax in a high level format to illustrate the basic concepts involved. For detailed syntax, refer to the Command Language Reference. For a detailed presentation on the RACDCERT command and to view some examples, please go on to Part II.

References

- **Security Server Manuals:**
  - **RACF Command Language Reference (SC22-7687)**
  - **RACF Security Administrator's Guide (SC22-7683)**
  - **RACF Callable Services Guide (SC22-7691)**
  - **LDAP Administration and Use (SC24-5923)**
- **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)**
- **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
- **PKI Services Red Book:**
  - http://www.redbooks.**ibm.com**/abstracts/sg246968.html
- **Other Sources:**
  - **PKIX - http://www.ietf.org/html.charters/pkix-charter.html**