Redbooks
ibm.com/redbooks

Managing Digital Certificates across the Enterprise

Keith Winnard

Martina von dem Bussche

Wai Choi

David Rossi

Security

z Systems

IBM

Redbooks

International Technical Support Organization

Managing Digital Certificates across the Enterprise

February 2016

SG24-8336-00

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (February 2016)

This edition applies to Version 2, Release 2, of IBM z/OS (product number 5650-ZOS).

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of 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.

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

DB2®
Domino®
IBM®
RACF®
Redbooks®
Redbooks (logo) ®
System z®
Tivoli®
WebSphere®
z Systems™
z/OS®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get personalized notifications of new content
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android

Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication is the first in a series of five books that relate to the implementation and management of digital certificates that are based on a public key infrastructure. Digital certificates play a major role in the protection of data communications and their use continues to grow.

This Redbooks publication includes the following chapters:

- Chapter 1, “Digital certificates overview” on page 1 provides an overview of digital certificates. It describes their purpose, gives a high-level overview of how they are created and their relationship to keys and encryption, and how they can be deployed into an organization.
- Chapter 2, “Digital certificate management considerations” on page 19 describes choices and their possible effects to consider for setting up and organizing the infrastructure and processes to be effective in your environments.
- Chapter 3, “Introducing z/OS PKI Services” on page 27 describes how the IBM z/OS® PKI services can provide you with a cross-platform solution to manage your digital certificates and build a strong solution that uses established qualities of service.

After you read this IBM Redbooks publication, we suggest that you progress to *z/OS PKI Services: Quick Set-up for Multiple CAs*, SG24-8337.

Your comments are appreciated. Your feedback can help improve the quality of our Redbooks publications so other readers can gain more value from them.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Keith Winnard is the z/OS Project Leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and is keen to engage with customers to understand what they want from IBM Redbooks Publications. Before joining the ITSO in 2014, Keith worked for clients and Business Partners in the UK and Europe in various technical and account management roles. He is experienced with blending and integrating new technologies into the traditional landscape of mainframes.

Martina vondem Bussche is an IT Security Architect at the Client Center in the IBM Research & Development lab in Germany. She is certified ISACA Information Security Manager (CISM) and Information Systems Auditor (CISA). Having started her career at IBM as a systems engineer in technical pre-sales for Mainframe hardware and continued in technical pre-sales for IBM System z® software security products, she has a strong Mainframe background. Now, she focuses on overall Mainframe security topics and projects.

Wai Choi is a Senior Software Engineer for IBM in Poughkeepsie. She works on digital certificate support in IBM RACF®, PKI services, and Kerberos. Wai actively participates in conferences and forums about digital certificate and related topics. She is a Certified Information Systems Security Professional (CISSP).

David Rossi is an IBM z Systems™ Security Architect in IBM North America. He has a bachelor of science in Chemistry from Ithaca College and an MBA from Rutgers University. David has been at IBM for 17 years always working to design new workloads that run on z/OS and Linux on z System. He also is active in promoting the z System academic initiative, z Systems Ecosystem, and is leader of the New York Area IBM Linux on z Systems Executive Advisory Customer Council

Thanks to the following people for their contributions to this project:

Bob Haimowitz (Development Support Team [DST], Poughkeepsie Center) for setting up and maintaining the systems, and providing valuable advice, guidance, and assistance throughout the creation of this IBM Redbooks publication.

Rich Conway (DST, Poughkeepsie Center) for setting up and maintaining the systems, and providing valuable advice, guidance, and assistance throughout the creation of this IBM Redbooks publication.

Jo Johnston (GTS, Mainframe Serviceline Architect, IBM UKI) for advice and guidance throughout creation of this book.

Alice Shitari (IBM) for advice and guidance throughout creation of this book.

Vincente Ranierie Jr (IBM) for advice and guidance throughout creation of this book.

Thank you to the following people for setting up and supporting the project and the residency:

Peter Bertolozzi (Systems Management specialist, IBM Redbooks residency support, Poughkeepsie Center) for setting up and maintaining the environments for residents to work in.

John Gierloff (Operations, Poughkeepsie Center) for residency set-up and support.

Don Brennan (DST, Poughkeepsie Center) for setting up and maintaining the systems hardware that was used in the creation of this IBM Redbooks publication.

Ella Buslovich (Graphics specialist, Poughkeepsie Center) for providing guidance and specialist graphics for this IBM Redbooks publication.

Ann Lund (ITSO Administration, Poughkeepsie Center) for administrative support to enable the residency.

Cheryl Gera (ITSO Administration, Poughkeepsie Center) for managing the business operations for this IBM Redbooks publication.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Digital certificates overview

In this chapter, we present a high-level view of the purpose of digital certificates and how they might be used.

This chapter includes the following topics:

1.1 Goal

The goal of this chapter is to help you gain a high-level understanding of digital certificates and how they might be used. Specifically, we describe the following topics:

- The need for commendations protection
- Keys, signatures, digests, and certificates
- Certificate Authorities

1.2 Overview

Digital certificates are at the heart of protecting all aspects of data communication, from websites for business and banking to shopping and product development, to social media for interaction and collaboration.

The information explosion, the adoption of cloud computing, and governmental regulations are making digital certificates even more important than ever.

Two parties are involved in the use of certificates (in this book, we refer to digital certificates as certificates). One party uses a certificate to identify itself, the other party must validate it. This process is referred to as a *handshake*. The protocol that is used is Secure Sockets Layer/Transport Level Security (SSL/TLS). For the handshake process to work, both parties must store the certificates in their own certificate store. The certificate store is also referred as a *keystore* or a *key database*.

But where do you obtain the certificates to put in the certificate store for the SSL/TLS protocol?

A certificate authority (CA) issues certificates. There are well-known CAs that sell certificates. There are internal CAs that issue certificates for their own enterprise. All certificates are created by using common standards. That is, no matter which CA sells you the certificate, no matter what CA and on what platform the certificate is created, it can be used by any application on any platform. Therefore, choosing the CA is an independent consideration of the platform on which the certificate is being used.

If you have applications running on z Systems, Linux, and Windows that need certificates to operate and you want to use an internal CA, you do not need to have a CA running on every one of the platforms.

What platform do you want to run your CA? Selecting the platform is based on the following key considerations:

- Security
- Availability
- Scalability
- Functionality

This book describes these topics to give you a high-level understanding of the considerations and options you have to implement an effective way of creating and managing digital certificates with cost and value in mind.

1.3 Primary question

Certificates have one important primary question to answer, as shown in Figure 1-1.

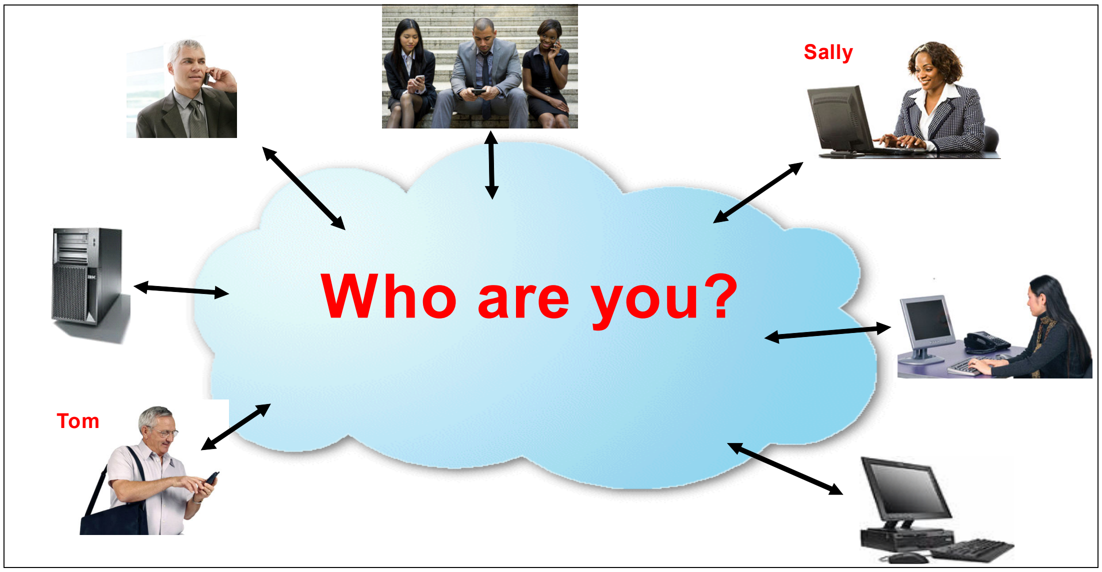


Figure 1-1 Primary question

Before communication starts, anyone or anything (such as a server) must make their identity known to others before proceeding with the communication process.

Note: Each party in the communication process is referred to as an *entity*; however, for the purposes of this overview, we refer to people communicating via mobile phones and notebooks (as shown in Figure 1-1) with Tom and Sally. This scenario can also apply to servers talking to each other or people and servers communicating with each other.

But is your identity really you? It might be someone posing as you.

Secondary questions

Tom might present his identity to Sally, but can she believe this is really Tom? Before we share any messages or transactions with anyone or anything, we must feel more secure about who is involved in the communication and determine whether they can be party to the communication. Three more questions are shown in Figure 1-2 on page 4.

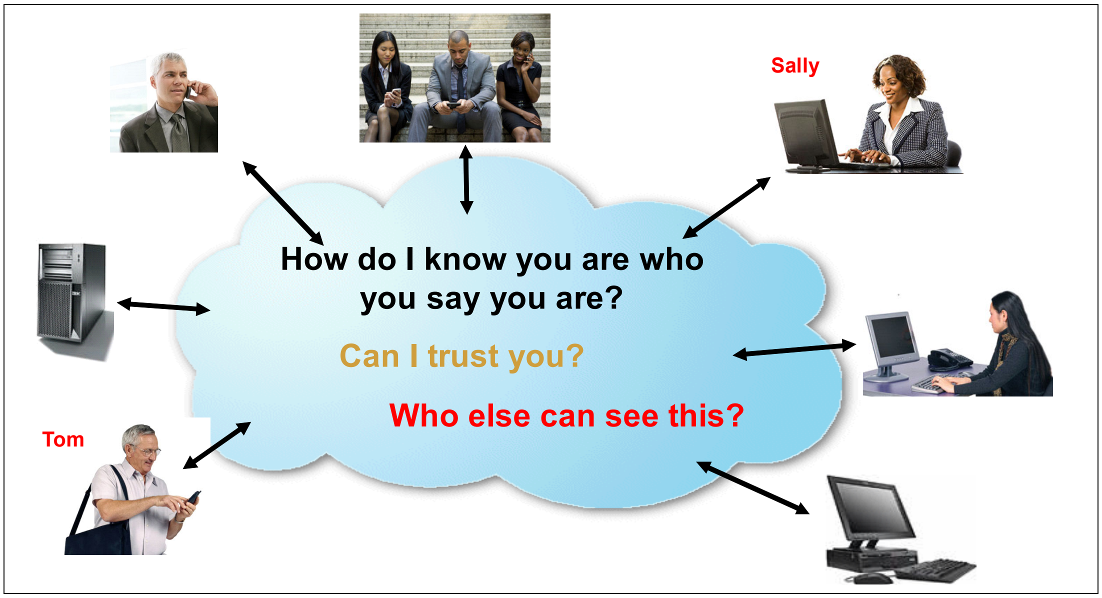


Figure 1-2 Secondary questions

Imagine Tom is making a purchase from Sally. Tom feels more comfortable about making a purchase if these secondary questions are answered before proceeding with the purchase, and has confidence in knowing with whom he is dealing.

The solution to the issues that are raised by our questions are certificates, as shown in Figure 1-3.

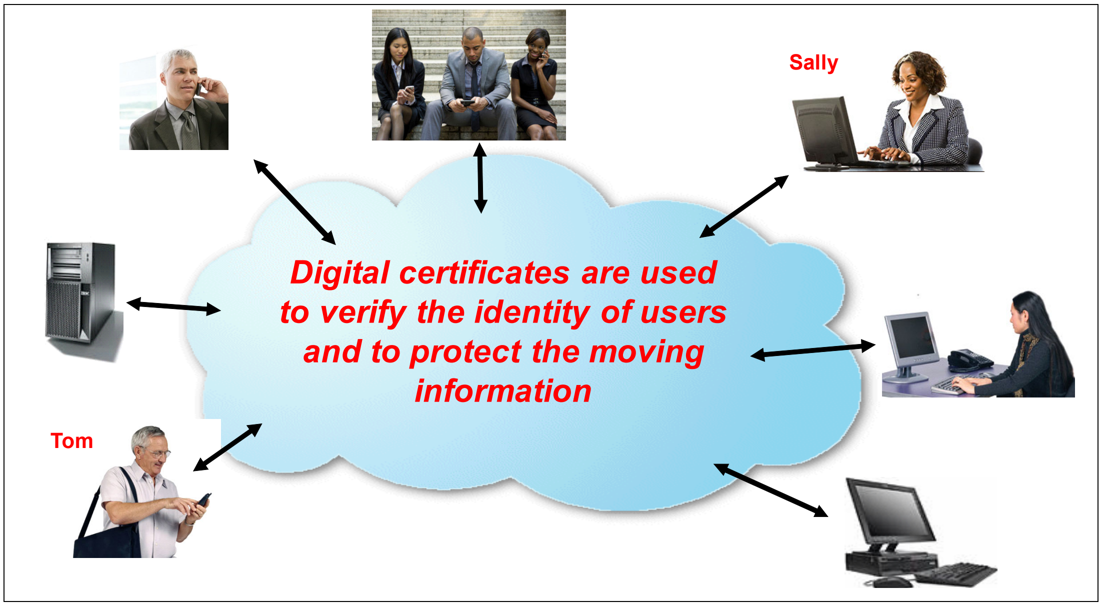

Sally
Digital certificates are used to verify the identity of users and to protect the moving information
Tom

Figure 1-3 Digital certificate's role

We can see in Figure 1-3 that if certificates can provide the authentication of people and servers that we want to communicate with, we can proceed with more confidence and an increased level of trust.

Are we okay with this process? Well, not quite. There is another aspect to consider, as shown in Figure 1-4.

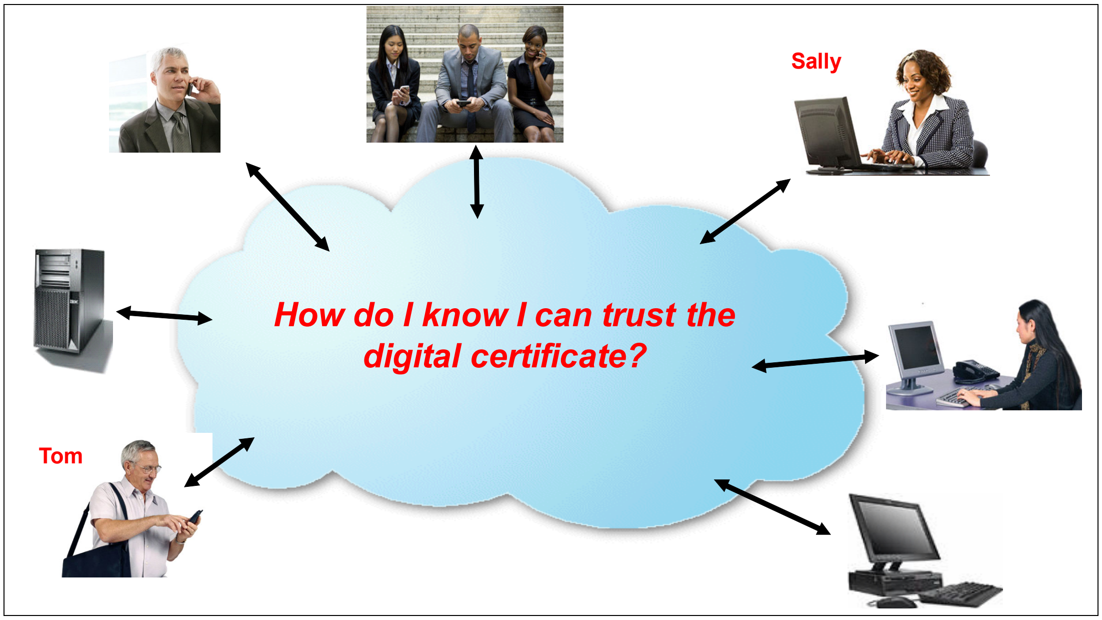


Figure 1-4 Can I trust a digital certificate?

In the same way you have concerns about identifying with whom you are communicating, the argument applies to the certificates themselves. We are facing the well-established question of “Quis custodiet ipsos custodes?” or, “Who will guard the guards?”

The solution is to establish a recognized way for people and servers to obtain a certificate that they can trust. The certificate must also provide information to be used to protect the messages and data during the communication process.

Note: The certificate does not perform any encryption. The information that is contained within it is used by other software and hardware to perform encryption.

The provision of a certificate is granted through a CA. In Figure 1-5, we see that a CA creates certificates for users.

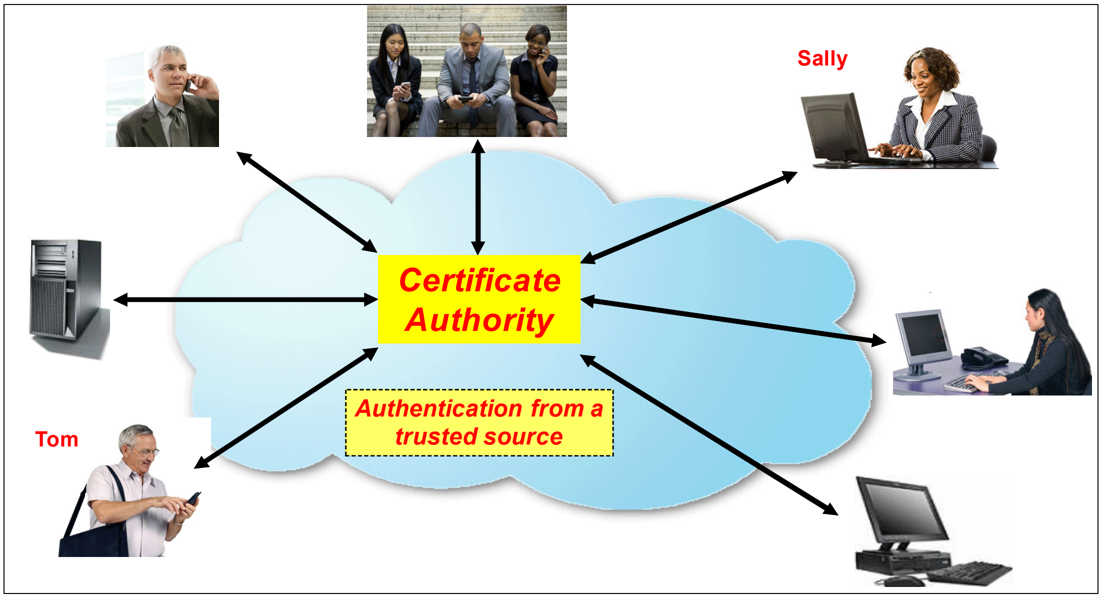
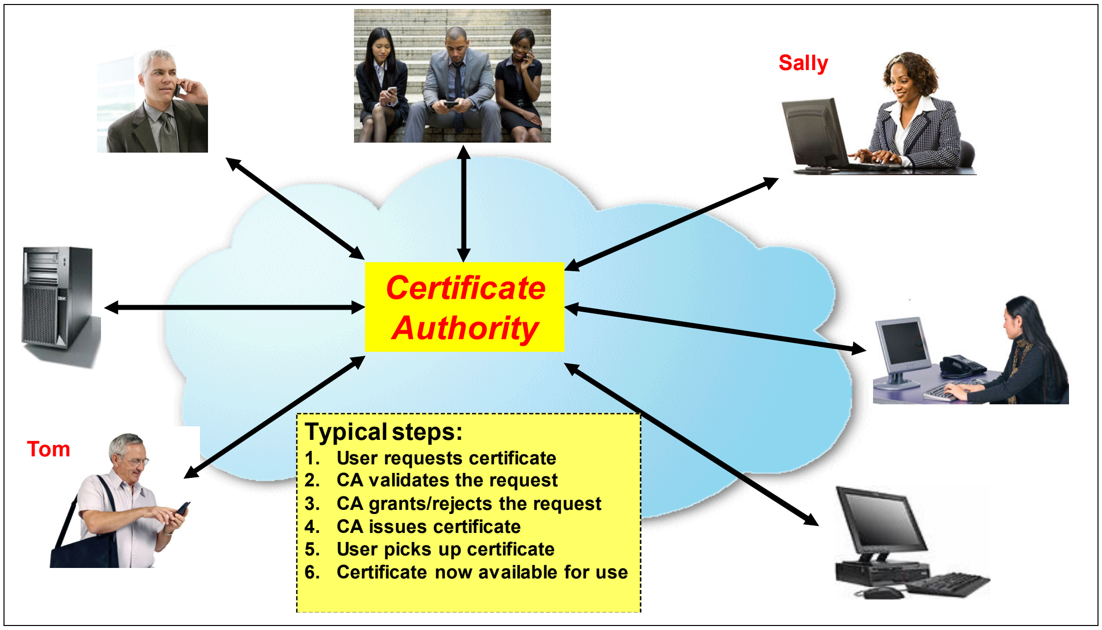


Figure 1-5 Certificate authority

The typical steps for this process to occur are shown in Figure 1-6.



Typical steps:

1. User requests certificate
2. CA validates the request
3. CA grants/rejects the request
4. CA issues certificate
5. User picks up certificate
6. Certificate now available for use

Figure 1-6 Creating a digital certificate

Although this overview is simplistic, section 1.4, "Behind the scenes" on page 7 describes varying levels of communication protection and what attributes must be met to help answer the questions posed thus far.

1.4 Behind the scenes

In this section, we describe the following topics:

- Security attributes
- Secret key cryptography
- Public key cryptography
- Digital certificates

We describe the basic information for each topic, and present a high-level understanding of how the different parts can work together to form a solution.

Security attributes

Communication includes the following essential security attributes:

- Authenticity: Verification of the validity of a person or server's claim to identifying who they or it is.
- Integrity: Verification that the message content was not changed or compromised.
- Confidentially: Ensuring that the message can be seen by the intended recipient only.
- Non-repudiation: Assurance that the sender or recipient cannot deny being the source or in receipt of the message.

Without these attributes, we are exposing our communications to a high level of risk and that exposure can result in unwanted consequences.

Therefore, how do we take steps to achieve a level of security that we can have confidence in?

Secret key cryptography

In our example, suppose that Tom wants to send a private message to Sally, who is another member of our group. As shown in Figure 1-7, Tom sends a message by using his mobile phone to Sally, who is using her notebook.

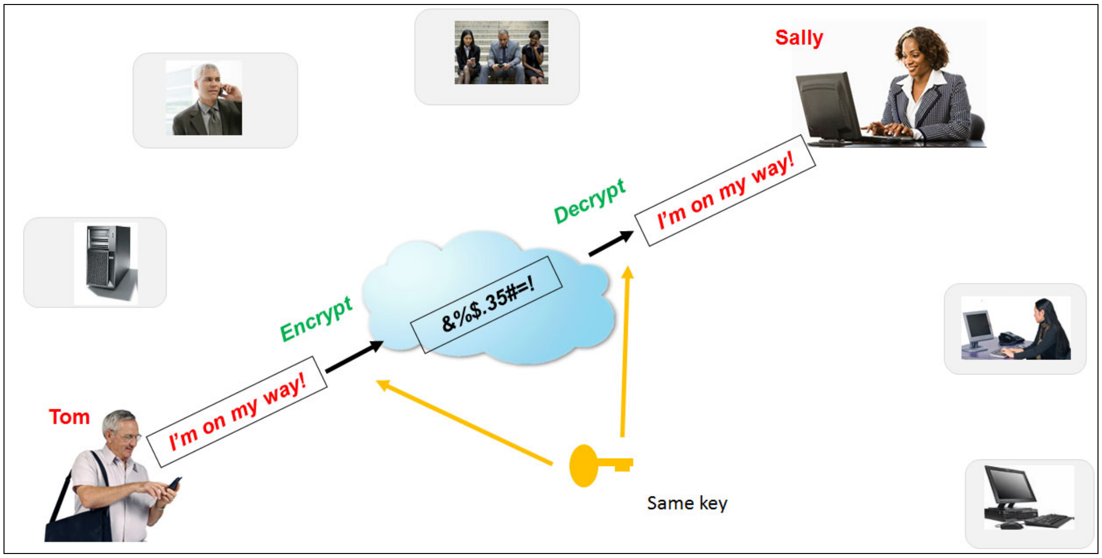


Figure 1-7 Secret key cryptography

They want the message to be private so only they can understand it. The following process occurs:

1. Tom types the message and sends it.
2. The message is encrypted by using a key.
3. When Sally receives the message, it is decrypted by using the same key.
4. Sally reads the original message.

In cryptography, an algorithm or set of algorithms use a key to encrypt the data. There are two parts: the algorithm and the key. These parts are separate from each other, and because they are separate from each other, a new key is required if the key is compromised but the algorithms can remain in place. There is no need to write a new algorithm; instead, use a different key.

In Tom and Sally's communication, the same key is used to encrypt and decrypt the original message.

The key is a *secret* between Tom the sender and Sally the receiver. If only Tom and Sally can access the key, consider the following points:

- If Tom uses this particular key to encrypt a message and send it to any other member of the group, the intended recipient does not have the key and cannot decrypt the message and read it.
- If Tom or Sally want to communicate with other members of the group, the following scenarios might occur:
  - Tom creates an individual key for every person and server in the group to communicate with each directly. In this case, he requires another six keys. If everyone in the group chose to communicate with each other by using individual keys, the number of keys would be 6+5+4+3+2+1 = 21. This number increases further if members of the group wanted to form sub groups that had their own keys to secure their communication.
  - Tom and Sally might choose to share the key with others. Those members who had access to the key can encrypt and decrypt messages with each other; however, the privacy level between the original Tom and Sally level now moves to a group level privacy whereby anyone who has access to the key can see the others' messages.

Note: In secret key cryptography, this type of key is called a *symmetric key*.

Public key cryptography

Public key cryptography consists of the following keys:

- Public key: This key is a key that is made available to the public.
- Private key: This key is kept private to the owner and not made available to the public.

The use of these two keys is known as a paired key approach. The key that is used for encryption is *different* from the key used for decryption. The public key has a counterpart called the private key. When the keys are created, they are created as a pair. This process is known as *asymmetric cryptography*, which is different from the secret key that is shown in Figure 1-7 on page 7 that uses a *s*ymmetric key (uses the same key for the encryption and decryption).

Asymmetric key pair features the following characteristics:

- An encrypted message can be decrypted only with the other key of the key pair.
- An encrypted message cannot be decrypted with the encryption key that was used to create the encryption.

- Each communication includes a public key and a private key. In our scenario, Tom wants to send a message to Sally by using the asymmetric key pairs. The following steps occur:
  - Tom accesses Sally's public key to encrypt the message.
  - Sally uses her private key to decrypt the message.

Figure 1-8 shows how public key cryptography handles the events that were shown in Figure 1-7 on page 7.

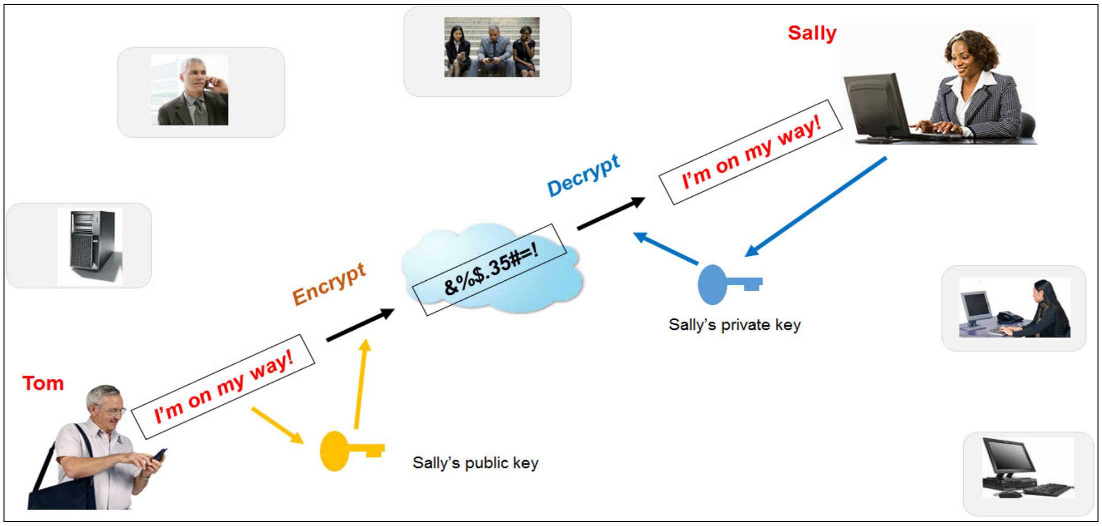


Figure 1-8 Public key cryptograph

Is this solution acceptable? Does it meet what we stated in "Security attributes" on page 7? No, it does not. It provided a level of confidentiality because the message was encrypted and remained so during its journey until Sally receives it and decrypts it. However, it did not provide sufficient protection for authenticity, integrity, and non-repudiation.

In Figure 1-8, Sally cannot be sure who the sender really is because the sender picked up her public key and used it. Tom did not supply any information that verifies his identity. Therefore, although Sally can decrypt his message and read it, she has no real proof of who sent the message. If Tom wants to prove only that the message is coming from him, he can use another technique: he can digitally sign the message.

Digital signatures

Figure 1-9 shows a different scenario for Tom to send his message to Sally.

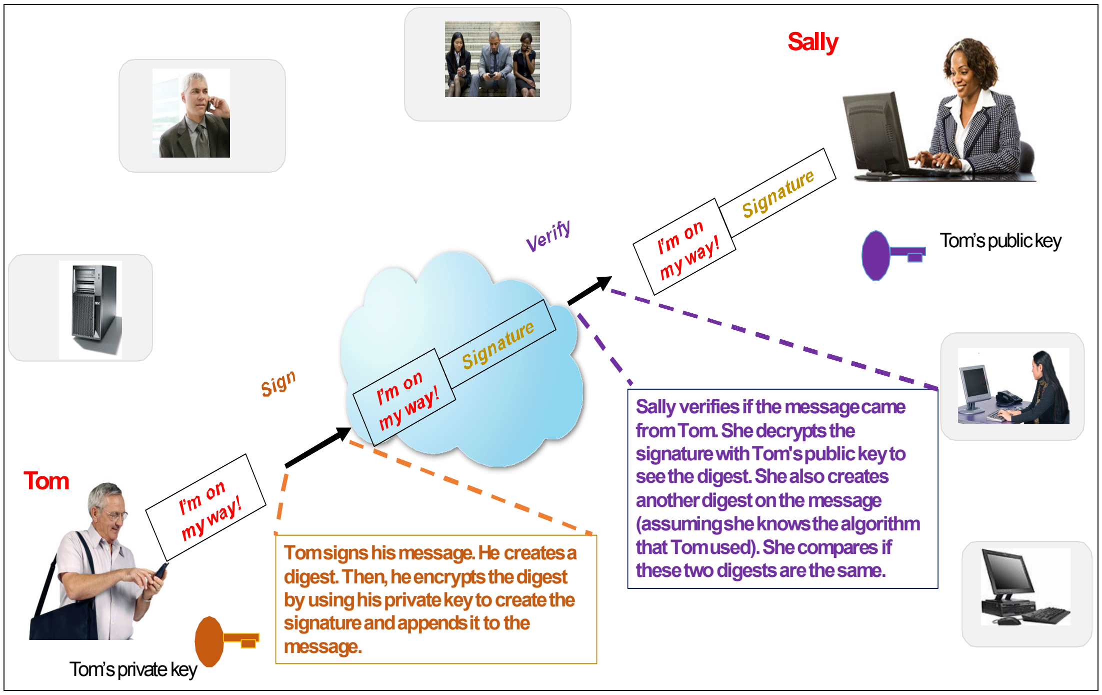

Sally
Tom's public key
Verify
I'm on my way!
Signature
Sign
Tom
Tom's private key
Tom signs his message. He creates a digest. Then, he encrypts the digest by using his private key to create the signature and appends it to the message.
Sally verifies if the message came from Tom. She decrypts the signature with Tom's public key to see the digest. She also creates another digest on the message (assuming she knows the algorithm that Tom used). She compares if these two digests are the same.

Figure 1-9 Digitally signing a message

This process includes the following steps:

1. Tom creates the message to send to Sally.
2. Tom uses an algorithm to hash the message. The hashed output is known as a *digest*.
3. Tom uses his private key to encrypt the digest to create a signature.
4. The signature is appended to the message and sent to Sally.
5. Sally uses Tom's public key to decrypt the signature. This decryption reveals the digest that Tom created.
6. Sally also received the original message, so she can use the same hashing process to create a digest.
7. If the message that Sally receives is the same the message that Tom sent, both of the digests should be the same if she uses the same hashing process. This process assures her that the message was not compromised.

This technique improves the situation. The digital signature that was provided by Tom using his private key can prove that the message is from him; otherwise, when Sally used his public key to decrypt the message, it does not work. The matching digests also prove that the message was not altered.

In the scenarios that describe the use of public and private key pairs thus far, the following fundamental questions still must be answered:

- How does Sally get Tom's public key?
- How does Sally know which algorithms to use to create the hash?

The answer to these questions is the digital certificate. This certificate is the container for the public key and the algorithm. It also indicates the owner of the certificate and other information.

However, the use of a certificate to carry the public key and the algorithm solves the problem partially. The problem moves to the certificate level in that how can Sally know that the certificate really belongs to Tom and that she can trust the content? If Sally cannot trust the issuer of Tom's certificate, she cannot trust the certificate.

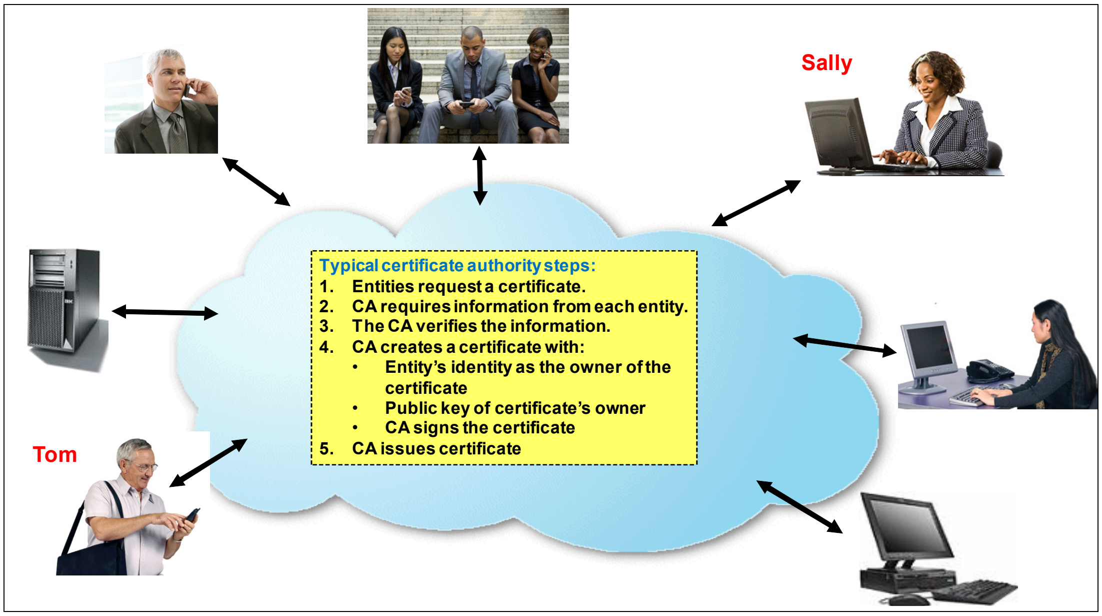
If a certificate is to be trusted, the creator and issuer of the certificate must be trustworthy. It is here that the concept of the CA is important.

1.5 Certificate authorities

The CA can issue digital certificates. The digital certificate must achieve the following goals:

- Verify that the person is who they say they are.
- Verify that the public key belongs to the person being verified.

The CA validates credentials of owner and signs the digital certificate. The CA's signature validates that the owner is who they say they are and that the public key belongs to them. The CA might also generate the key pair if the person or server has not already done so, as shown in Figure 1-10.



Typical certificate authority steps:

1. Entities request a certificate.
2. CA requires information from each entity.
3. The CA verifies the information.
4. CA creates a certificate with:
   - Entity's identity as the owner of the certificate
   - Public key of certificate's owner
   - CA signs the certificate
5. CA issues certificate

Figure 1-10 CA digital certificate creation and issuance

The CA must be a *trusted* third party because without this trust, the purpose of the certificate is defeated and we are still left with the issue of not knowing with certainty if the public key belongs to the sender. Figure 1-10 shows the following typical process to issue a certificate:

1. The CA collects and validates credentials from the certificate requester.
2. The CA accepts the request with the public key that was generated by the owner. If the owner wants the CA to generate the keys, it generates the public key and private pair.

3. The CA signs the certificate by hashing the content of the certificate, which includes the requester's public key to produce a digest followed by encrypting the digest with the CA's private key.
4. The signed digital certificate is available to the requester for pickup.

The certificate that is issued to the requester contains key fields, such as the owner's name, public key, issuers name, issuer signature, and the period of validity. There can be other information contained, which might be supplied at request time that can then be included in the certificate, as shown in Figure 1-11.

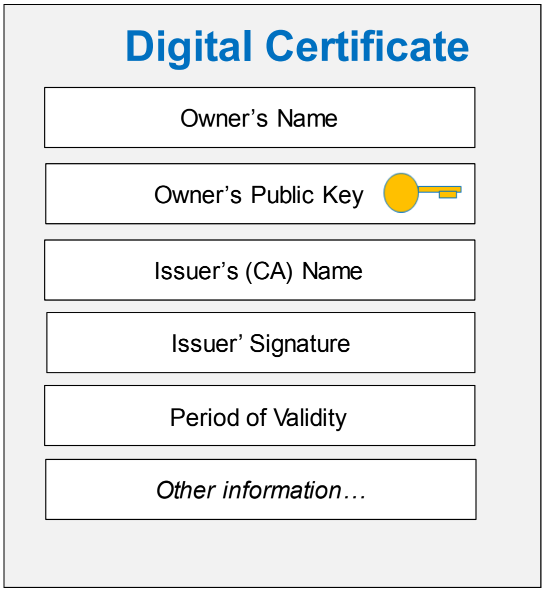


Figure 1-11 Sample digital certificate format

The amount of detail can vary because there is a demand for more than one type of certificate. That is, certificates can be created for different purposes; for example, email and VPN.

The X.509 standard introduced extensions to digital certificates that cater for the differing requirements but still remain true to the one type of digital certificate.

In our scenario, the following sequence of events now occurs:

1. Tom attaches the certificate to his message, he encrypted it by using his private key.
2. Sally receives the message and accesses the certificate to complete the following tasks:
   a. Identify who is the owner of the certificate and message.
   b. Check the CA's signature on the certificate, and continues if she trusts that CA.
   c. Use Tom's public key to decrypt the message.

We have one last matter to address: How can the CA's identity be represented by its own certificate?

The CA must establish its own identity and credentials by taking the following actions:

1. Generating its own key pair.
2. Protecting its private key so it is not available to entities.
3. Creating its own certificate as being the owner and including its public key.
4. Self-signing the certificate.

The CA is now available to receive requests and to authenticate its requesters.

1.5.1 Other checks for validating a digital certificate

Before going through the decryption and comparison processes, the following checks must be made:

- If the certificate expired.
- If the certificate was revoked.

If the digital certificate expired or was revoked, it is not valid and cannot be trusted.

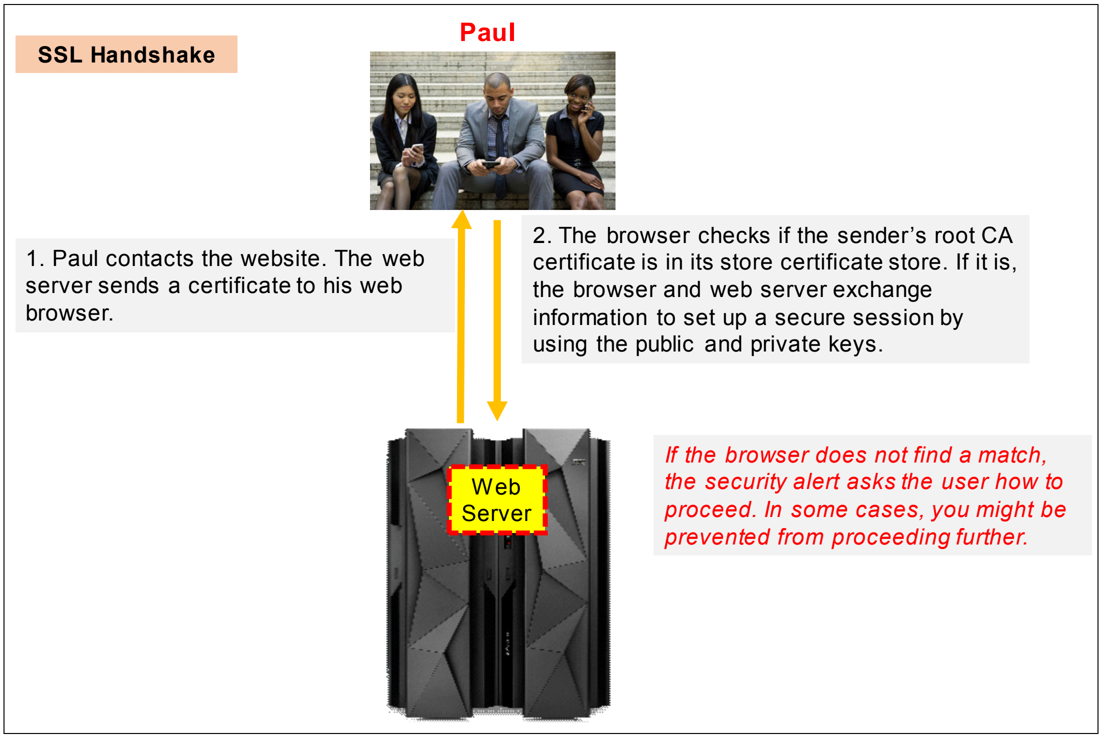
1.6 Authentication versus authorization

There is a myth that a digital certificate provides a level of authorization. This is not the case.

> **Important:** A certificate is used only to verify identity. The receiving system can map the identity to authorization credentials, but the certificate does not contain authorization credentials.

1.7 Use of a digital certificate

A common use of digital certificates is between a web browser and a web server, as shown in Figure 1-12.



1. Paul contacts the website. The web server sends a certificate to his web browser.

2. The browser checks if the sender's root CA certificate is in its store certificate store. If it is, the browser and web server exchange information to set up a secure session by using the public and private keys.

If the browser does not find a match, the security alert asks the user how to proceed. In some cases, you might be prevented from proceeding further.

Figure 1-12 SSL handshake between browser and web server

Paul is using the browser on his phone to access a website. For him to validate that the web server is who it says that it is, a certificate is sent to the browser. The browser has a trusted certificate store. If the issuer or CA of the certificate is known and trusted, the browser and web server exchange information to set up a secure session by using the public and private keys. This process is known as the Secure Sockets Layer (SSL) handshake. If the browser does not find a certificate match, a security alert prompts the user about how to proceed. There are circumstances where some browsers refuse to allow you continue any further.

SSL handshake and SSL protocol clarification

We used the term *SSL handshake* and not the term *SSL protocol*. To avoid confusion and misuse of the terms, consider the following points:

- An SSL handshake proves who we are.
- An SSL protocol is the language that we speak to communicate with each other.

1.7.1 Integrating certificates into applications

Suppose that a new business application must communicate with another application. We might want to encrypt the communication through the SSL/TLS protocol, as shown in Figure 1-13. Certificates and CAs are platform-independent. Some of the CAs that are shown might be well-known and trusted, but others might be private CAs within an organization. However, if they all use the SSL/TLS protocol, they can communicate with each other after their identity is proven by the SSL handshake.

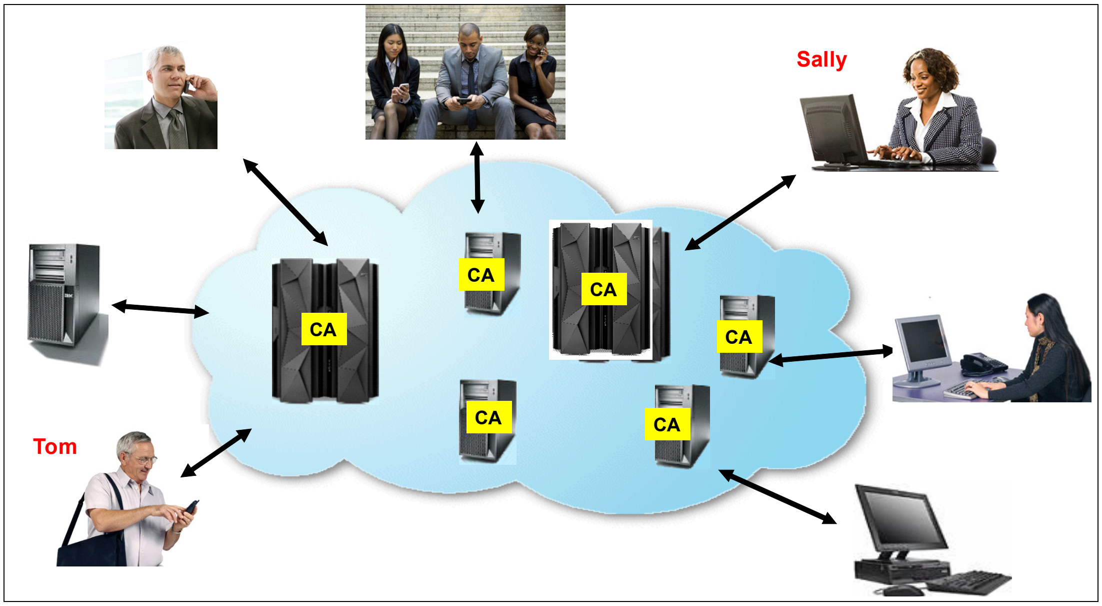


Figure 1-13 Cross platform business application

Before implementing a project that requires data communications, the interested parties that are involved with each part of the application must agree on a collection of certificates and keys that are needed to provide secure communication. The agreed collections must allow for the development, testing, and implementation phases of the project. This process most likely involves setting up a collection of keys and certificates into a various stores.

If the applications are large, it is also likely various certificates are required for the individual areas of the application. These certificates require the appropriate keys.

Organizing the collections to make them effective involves the following tasks:

- Identifying the location of the keystore
- Determining which certificates are needed and where they are needed

Stage 1: Identifying the location of the keystore

Examine the application and see how it is configured to identify the best places to store the keys (there might be more than one place). Define which configurations are to be used and determine their location, as shown in Figure 1-14. Each server has its own configuration and contains information or pointers to locations, such as keystores.

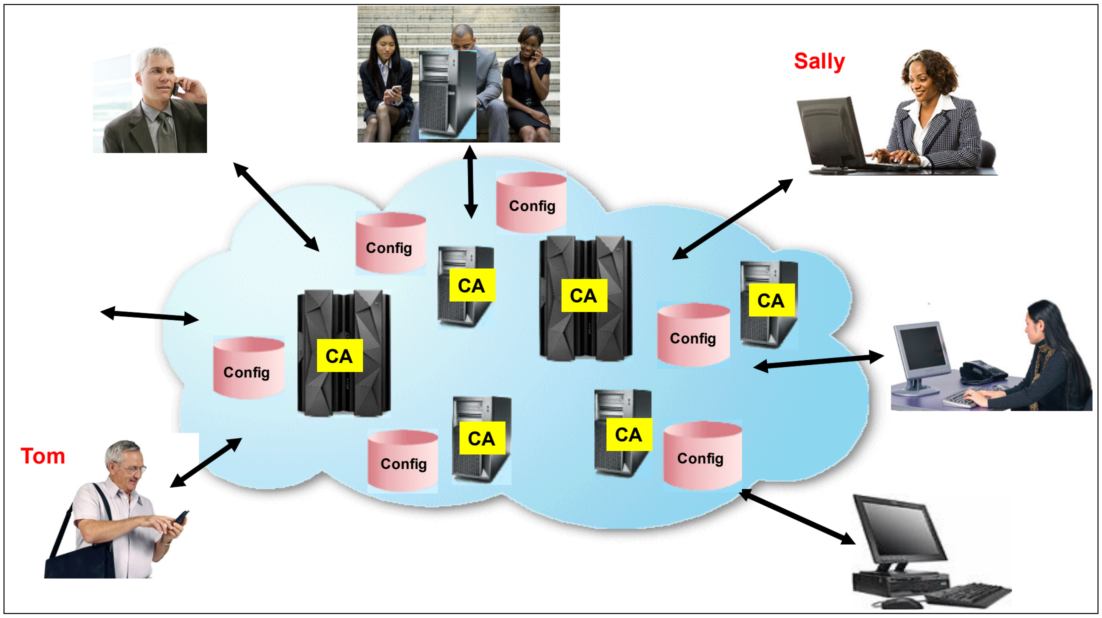


Figure 1-14 Configuring where the keystores are to be stored

Stage 2: Determining which certificates are needed and where

Examine all parts of the applications and based on the function and configuration that is identified in Stage 1, store the keys and certificates in the appropriate repositories.

Server authentication

The client starts the communication and attempts to establish communication with the server. The server must identify itself to the client. This process highlights why stage 1 and 2 are important because they ensure that the appropriate certificates are presented and the keys are available. Based on this exchange, the client can validate the server.

Client authentication

Having identified itself to the client, the server might now require that the client identify itself so that the server can validate the client.

1.7.2 Intermediate CAs

CAs can be hierarchical, which is useful for providing granular authentication. For example, an application might feature email, a document repository, and room booking functions. Therefore, these features are sub-applications and might have corresponding CAs to manage the certification for each area. So, our business applications integration can now be much more hierarchical, as shown in Figure 1-15.

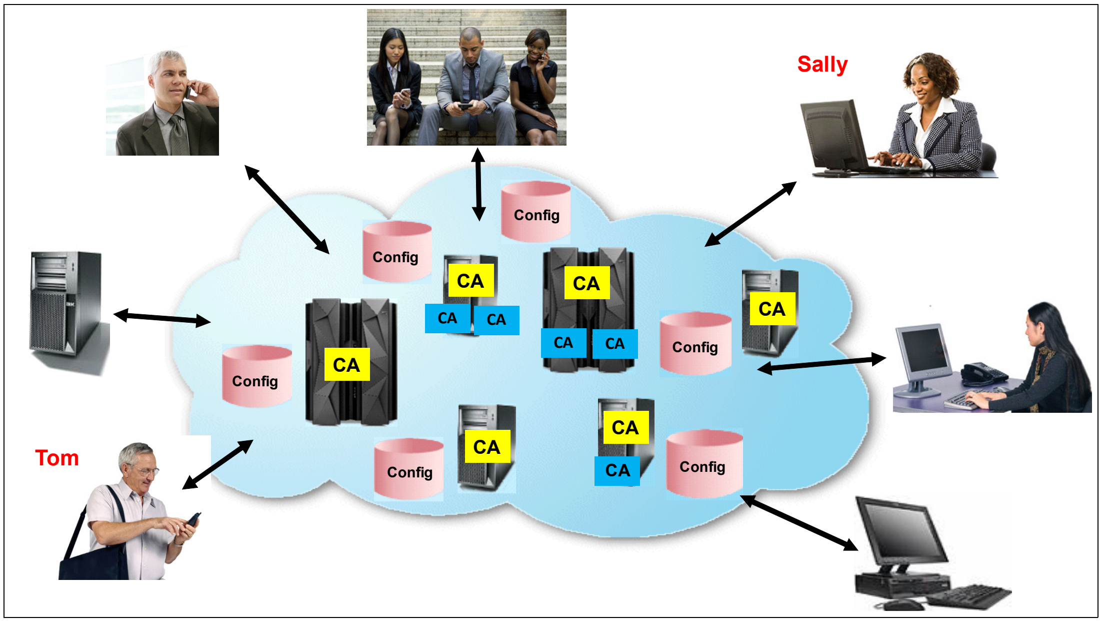


Figure 1-15 Intermediate CAs to support business applications

Note: The top-level CA is known as the *root CA*. If the browser trusts the root CA, it automatically trusts any intermediate CA that belongs to the root CA. This configuration is often referred to as the *chain of trust*.

To help with the structure to make the management and administration of all the CAs easier and consistent, one CA can be the root CA of all the other intermediate CAs. Your structure might not link to all CAs as the others might be owned by an external third party or be on different platforms. The CAs might not be connected but are still part of the application, as shown in Figure 1-16.

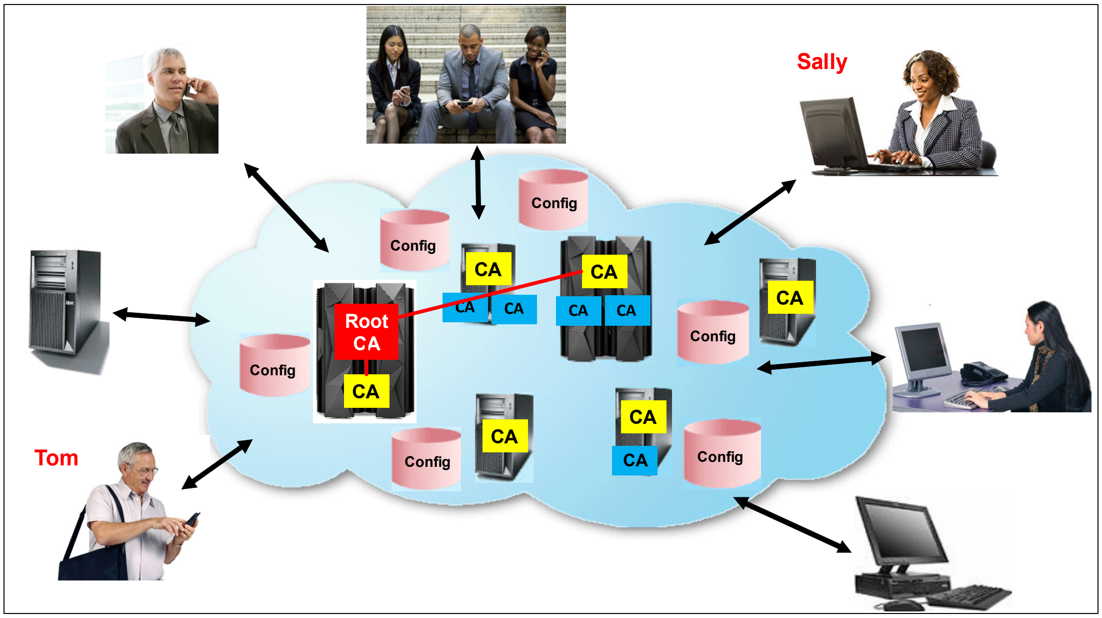


Figure 1-16 Easing the management and administration of CAs

The topics that we described in this chapter that relate to certificates and asymmetric keys form the foundations of a Public Key Infrastructure (PKI). For more information, see Chapter 3, “Introducing z/OS PKI Services” on page 27.

2

Digital certificate management considerations

This chapter provides an overview of topics for you to consider for managing your digital certificate lifecycle.

This chapter includes the following topics:

2.1 Goals

In this chapter, we have the following goals:

- Review the digital certificate lifecycle
- Consider how to manage the digital lifecycle

2.2 Using internal or external certificate authorities

Different aspects must be examined to decide whether to use an internal or external certificate authorities (CA) to create and manage digital certificates. The topics in this section can help you to think about which certificates are better-suited to internal or external certification.

An internal CA is installed and maintained within your enterprise, issuing certificates for the people and servers of your internal IT system.

An external CA is installed and maintained outside of your enterprise and is treated as a third party. Certificates are purchased from a well-known CA.

2.2.1 Costs

Buying digital certificates from a well-known CA creates financial costs. Our research at the time of this writing shows that prices can vary 20 - 1,200 US dollars per SSL certificate for the first year, depending on your requirements and the CA that you choose.

Although it is suggested to use certificates from a well-known CA (for example for external or public-facing web pages), it might not be necessary for internal (or private) IT systems. For internal IT systems or communication among a closed group of entities, a private CA can be used. Issuing certificates from within your organization avoids the costs of purchasing them from an external source.

2.2.2 Scope of use

The choice of choosing an internal or external CA can depend on the scope of use. Public CAs generate digital certificates for the public; that is, for anyone. For internal IT systems, you might want to make sure that only a closed group of entities can access them.

An analogy can be made here between internal and external certification. Next, we consider two items of identification, a passport and a company JOB ID card, as shown in Figure 2-1 on page 21.

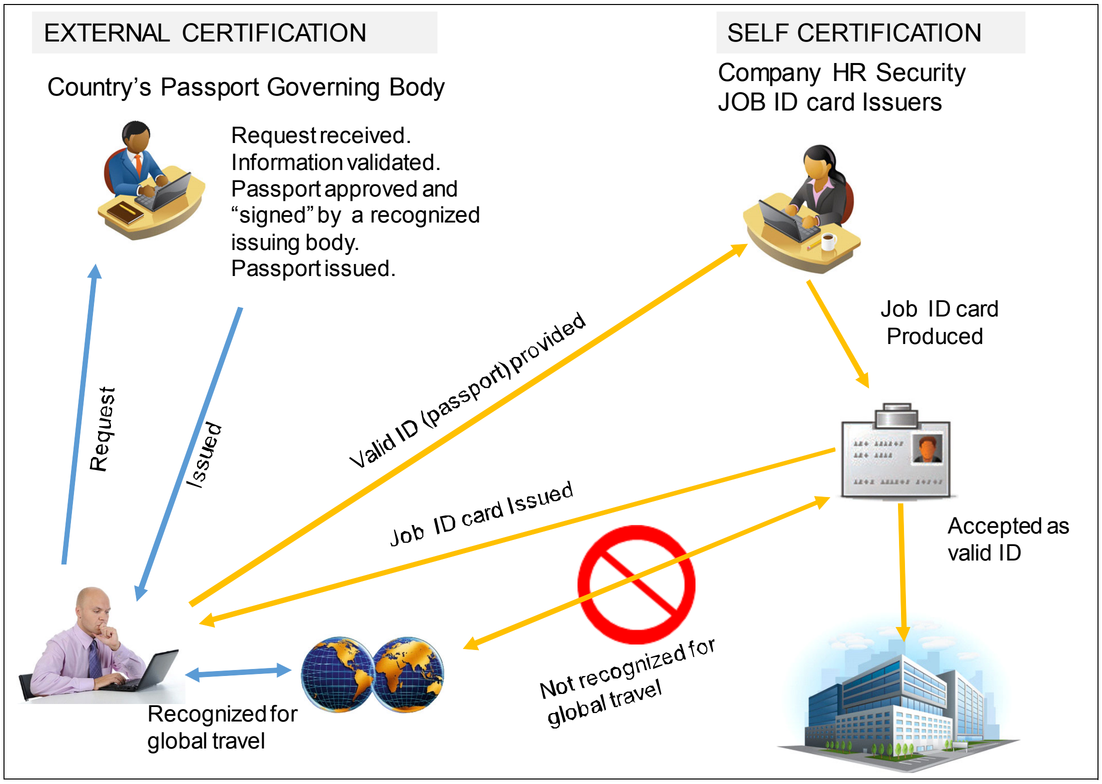


Figure 2-1 External and self-certification

The passport is externally validated and issued and provides the carrier with recognized global identity authentication whereas the Job ID card is accepted only within the company it was issued. It is *not* accepted as global identity authentication as is the passport. However, the passport is accepted as an acceptable identity authentication for the company to issue a job ID card.

2.2.3 Expanding scope of use

The certificates that are issued by internal CAs can expand beyond the organization to which the CA belongs. The internal CA can issue certificates for the organization's Business Partners, as shown in Figure 2-2 on page 22.

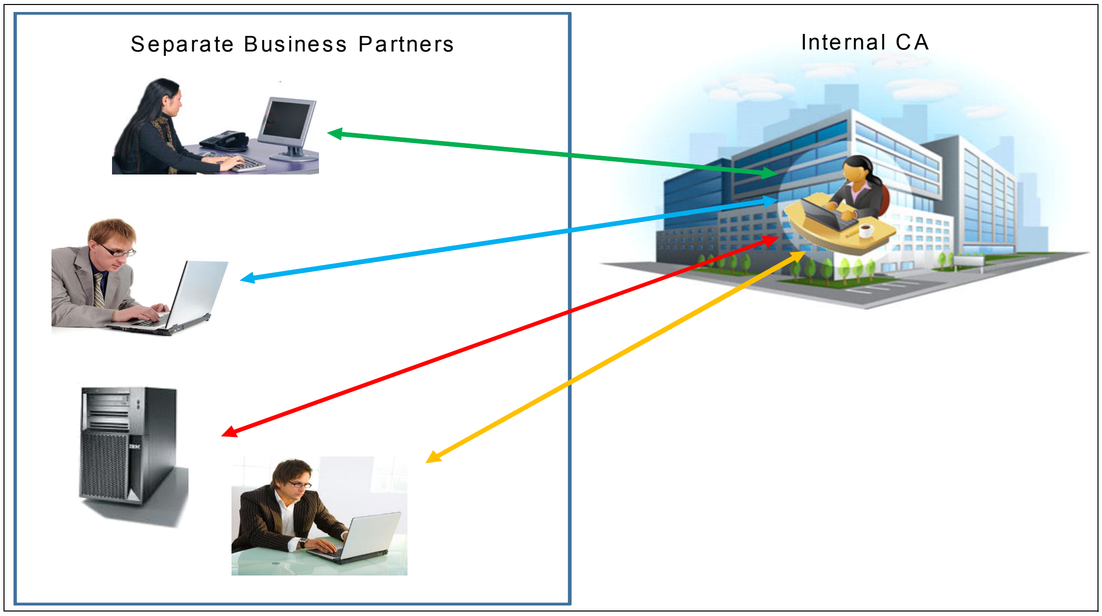


Figure 2-2 Internal CA creating certificates for Business Partners

The Business Partners might need certificates to deal with your organization. If your organization has an internal CA, your Business Partners must we willing to accept your CA certificate into their certificate store. This difference is a subtle change because your internal CA is now dealing with external entities. The effort and administration that is involved must be balanced against the costs and convenience of buying the certificates from a well-known CA.

The CA's certificate must be in each Business Partners certificate store, which can become a high administration effort as the number of Business Partners increases. Conversely, well-known CA certificates are included in the certificate store.

2.2.4 Compromised certificate considerations

Compromised certificates must be revoked. If a certificate is compromised, the issuing CA must revoke it and a request for a new certificate must be made. The requester can then use the new certificate instead of the compromised certificate.

If an intermediate CA is compromised, the root CA can revoke the intermediate's certificate. All of the certificates in the intermediate CA's chain are unusable. The root CA can issue a new certificate to the intermediate CA.

If a well-known company that provides certificates to high-profile clients (such as, financial institutions and major retailers) has its root CA compromised, all of the certificates it issued are unusable. This situation is serious and has visible affect on trading, which can make headline news.

2.3 Digital certificate lifecycle

Digital certificates go through phases. There are key activity points within these phases, which are referred to as the *digital certificate lifecycle*.

2.3.1 Lifecycle management

The lifecycle of a certificate requires managing. The following points are typical activities within the certificate's lifecycle:

- Request or Renewal: The initial request for the digital certificate or a renewal when the certificate is to expire.
- Approval: The request is approved or rejected based on the request information.
- Generation and Distribution: The digital certificate is generated with all the required elements and is distributed.
- Usage: The usage period occurs.
- The digital certificate expires and the end of the requested usage period or it might be revoked by an administrator for any valid reason, such as the keys being compromised.

Figure 2-3 shows the lifecycle of a certificate.

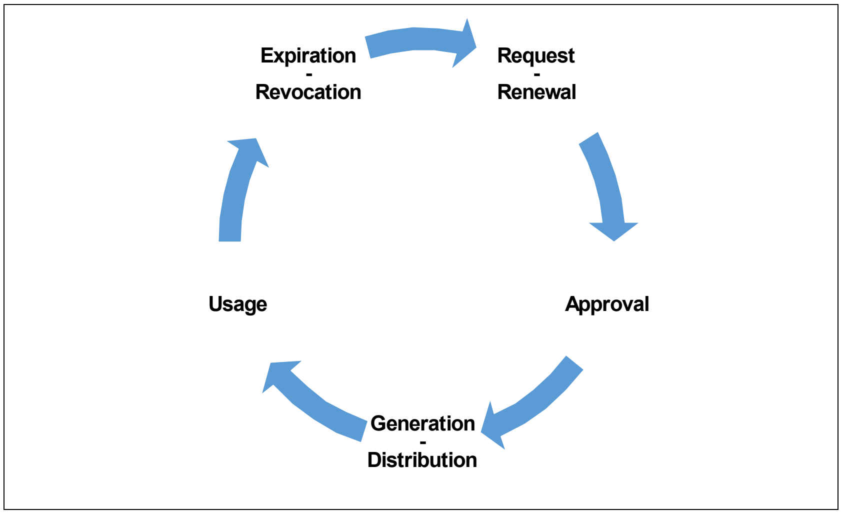


Figure 2-3 Digital certificate lifecycle

You can think of a digital certificate as a passport. It is used to prove that you are the person you say that you are and it is coming from a trusted authority: the government. This analogy also can be used to explain a certificate lifecycle. A certificate lifecycle includes the following steps:

1. As with a passport proving your identity, a digital certificate must be requested at a trusted authority (CA).
2. Someone at the authority must prove the request and approve or reject the certificate request.
3. After a certificate request is approved, the certificate can be generated and made available for distribution.
4. The certificate is used to authenticate an entity at a certain IT service or used to secure a communication. (For more information about the various use cases for certificates, see Chapter 1, "Digital certificates overview" on page 1).
5. The expiration date depends on the initial parameters in the certificate request. After a certificate expires, it is no longer usable. Revocation also can stop a certificate from being used even though it did not expire. It is the responsibility of the CA to post the revocation list. Entities accessing the certificate must check to see whether the certificate was revoked before trusting it.

6. After expiration, a certificate must be renewed. After revocation, a new certificate must be reissued.

2.4 Digital certificate lifecycle management considerations

In this section, we describe the areas for you to consider when developing your management strategy and processes.

2.4.1 Manual management risk

The increasing volume of certificates affects how you can choose to manage the lifecycle of certificates. Consider the following points relating to the manual management approach:

- Generation: Increased volume of certificates makes it harder to handle manually because it is time-consuming and requires extra manpower and training to satisfy the increasing demands.
- Distribution of certificates: Each certificate must be individually distributed. It can be more efficient to allow entities to retrieve certificates via download or standard protocols.
- Errors: Manual systems tend to be more error prone and subject to human limitations.
- Expiration: A growing challenge is to track expiration dates and to protect against the consequences of authentication failure because of certificate expiration.
- Responsiveness: Requesters must set the correct expectations that other actions can later depend upon the authentication being there; for example, when introducing new functions or creating an application.

  Certificates can be required throughout each stage of the application development, testing, and implementation lifecycle. Delays in the generation can affect subsequent activities. An automated process helps in defining a Service Level Agreement (SLA) for the delivery points of the management cycle.

Automated procedures help resolve many of the issues of manual management.

2.4.2 Request and approval policy

Roles and user autonomy are two considerations to help in defining the request and approval policy.

Roles

Depending on the internal IT security policies, the following questions must be asked:

- Who in an enterprise is allowed to request certificates?
- Who or how many administrators must approve a certificate request?
- Is auto-approval to be allowed, and to what level?
- Is there an audit trail for auto-approval or any other approval process?

The requirements of the various IT security policies can be met by using a public key infrastructure (PKI) and establishing a consistent and accountable workflow for the certificate request and approval process.

User autonomy

You might need to define a structure for user rights management, whereby personal users (such as company employees) can request, revoke, and renew their personal digital certificates. However, they might not be allowed to request server certificates unless they also are server administrators and their job requires them to do so.

2.4.3 Expiration

An issue with certificates is that they expire. If a certificate expires, the authentication process no longer works because the verification of the certificates fails.

The effect of a failed level of authentication often causes disruption to applications or in more extreme circumstances can lead to outages of IT services. For example, authenticating failure in an online business banking application can cause problems that require immediate attention and a high priority resolution.

2.4.4 Revocation

A digital certificate can be revoked for the following reasons:

- Key compromise
- CA compromise
- Affiliation changed
- Superseded
- Cessation of operation

2.5 Accountability

Accountability is key to demonstrate to auditors that certificates are managed across the enterprise in accordance with the enterprise's defined policies. The accountability is essential to verify that processes that relate to the digital certification lifecycle can be monitored to detect anomalies.

In addition, quantifiable accountability and monitoring can be used to measure and report activity levels that relate to the lifecycle and to use trending to predict activity levels within the lifecycle.

2.6 Public Key Infrastructure

Thus far we reviewed the use of keys and encryption, the role and structure of certificates, and the CA. We also described considerations that we can make to engage the appropriate structure, roles and responsibilities, options, and the potential activities that are involved throughout the enterprise or organization and across different kinds of servers and various operating systems. If we put all this information together into a viable, effective, and trustworthy solution, we have what is referred to as a PKI.

A PKI is based on public key cryptography. The infrastructure can consist of hardware, software, and policies to create, manage, store, distribute, and verify digital certificates.

Managing certificates centrally in a PKI helps you track all the activities that occur during the lifecycle of a certificate; for example, who requested a certificate and who approved it.

2.7 Regulatory demands

The regulatory requirements for digital certificate-related processes vary depending on the following factors:

- Country or government policy
- State within that country
- Type of business or organization:
  - Financial
  - Healthcare
  - Social welfare
  - Travel and accommodation
- Online culture and openness

Regulatory recommendations or requirements regarding the use of a PKI to manage digital certificates include the following examples:

- bsi Grundschutz (Germany) recommends the use of PKI for the following use cases:
  - Email encryption or signing
  - Remote access to company network (VPN) and IPSec
  - Authentication to wireless LAN
  - Signing code
  - Signing XML in web services calls
- HIPAA recommends the use of a PKI as the “most viable technology that will ensure the proper level of protection for healthcare information”.

Several countries are introducing laws or regulations around electronic signatures that require the use of a PKI.

3

Introducing z/OS PKI Services

This chapter gives you a brief overview of what z/OS PKI Services can provide to accommodate the management considerations and shows how flexible it can accommodate the certificate lifecycle.

This chapter includes the following topics:

- 3.1, “Goals” on page 28
- 3.2, “z/OS PKI Services functions” on page 28
- 3.3, “z/OS PKI Services elements overview” on page 33
- 3.4, “Added value of z/OS PKI Services” on page 39
- 3.5, “Certificates across the enterprise” on page 41
- 3.6, “What is next?” on page 42

3.1 Goals

In this chapter, we have the following goals:

- Show how the z/OS PKI Services functions fit into the digital certificate lifecycle.
- Clarify the similarities and differences of the administrator and requester (or user).
- Provide an overview of the z/OS PKI Services elements.

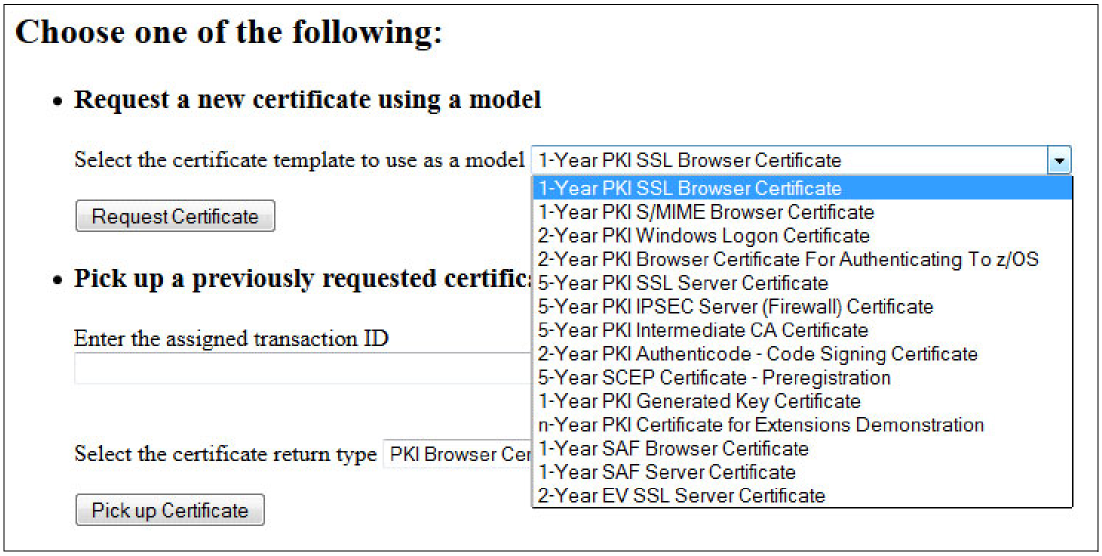
3.2 z/OS PKI Services functions

The z/OS PKI Services are structured to help you manage your organization and help simplify some of the procedures and processes of the certificate's lifecycle. z/OS PKI Services includes functions for the following roles:

- The certificate requester, which can be a person or a server (sometimes referred to as the user).
- The administrator or team of administrators.

3.2.1 Certificate templates

z/OS PKI services provides you with default certificate templates within the web application. You can request different kinds of certificates that are based on the intended purpose and other considerations, such as the validity period. These templates can be customized and completed with default values. You also can indicate fields that are mandatory or optional to complete on the certificate request form, as shown in Figure 3-1.



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

- 1-Year PKI SSL Browser Certificate
- 1-Year PKI S/MIME Browser Certificate
- 2-Year PKI Windows Logon Certificate
- 2-Year PKI Browser Certificate For Authenticating To z/OS
- 5-Year PKI SSL Server Certificate
- 5-Year PKI IPSEC Server (Firewall) Certificate
- 5-Year PKI Intermediate CA Certificate
- 2-Year PKI Authenticode - Code Signing Certificate
- 5-Year SCEP Certificate - Preregistration
- 1-Year PKI Generated Key Certificate
- n-Year PKI Certificate for Extensions Demonstration
- 1-Year SAF Browser Certificate
- 1-Year SAF Server Certificate
- 2-Year EV SSL Server Certificate

Request Certificate

- **Pick up a previously requested certific**

Enter the assigned transaction ID

Select the certificate return type: PKI Browser Cer

Pick up Certificate

Figure 3-1 Use of the templates

Use of the provided certificate templates can ensure that all certificates that are requested by using your customized templates are aligned with your organization's IT security policies.

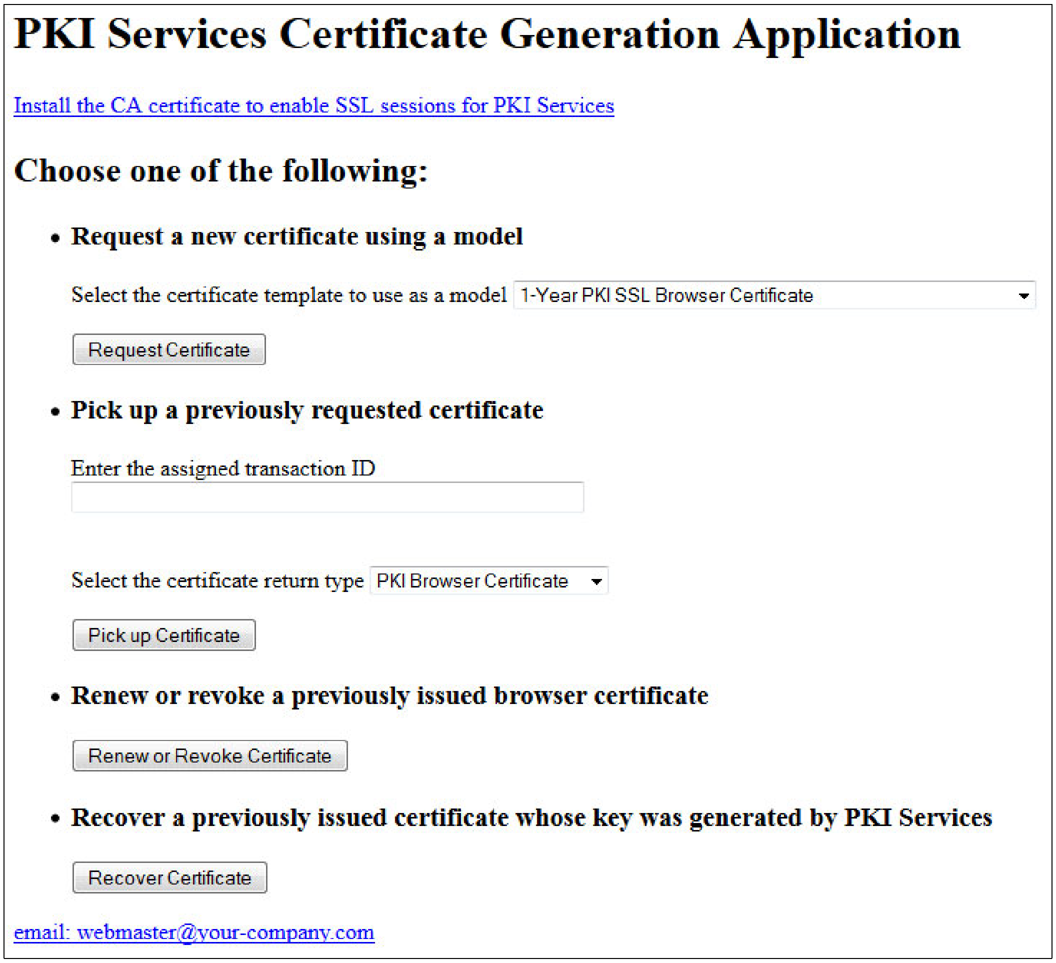
z/OS PKI provides the following services:

- Request or renew certificates
- Approve certificate requests
- Email notification

- Generate certificates
- Distribute certificates
- Revoke certificates

3.2.2 Requesting or renewing certificates

To request or renew digital certificates, z/OS PKI Services provides a user web application. This application guides the user to request and renew certificates through their web browsers. The application includes sample windows that can be easily customized to meet your organization's requirements, standards, and appearance, as shown in Figure 3-2.



PKI Services Certificate Generation Application

Install the CA certificate to enable SSL sessions for PKI Services

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

  Request Certificate

- Pick up a previously requested certificate

  Enter the assigned transaction ID

  Select the certificate return type: PKI Browser Certificate

  Pick up Certificate

- Renew or revoke a previously issued browser certificate

  Renew or Revoke Certificate

- Recover a previously issued certificate whose key was generated by PKI Services

  Recover Certificate

email: webmaster@your-company.com

Figure 3-2 User web application sample window

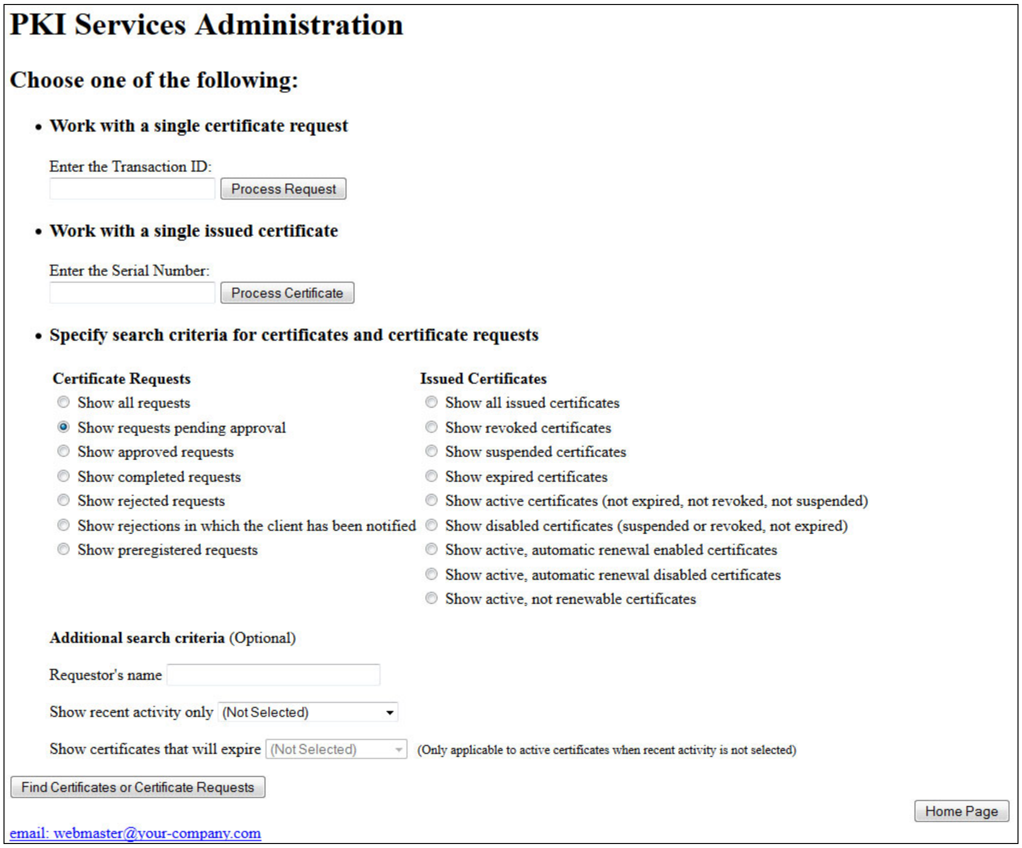
z/OS PKI services creates certificates for requesters who did and did not generate their own key pair. For those requesters who did not generate their own key pair, the key pair might be generated. However, the Integrated Cryptographic Service Facility (ICSF) must be configured to use this function.

3.2.3 Approving certificate requests

z/OS PKI Services provides administrative functions to approve, approve with modifications, or reject requests. It also supports the following tasks:

- Review pending certificate requests.
- Query pending requests to process those requests that meet certain criteria.
- Display detailed information about a certificate or request.
- Annotate the reason for an administrative action.

z/OS PKI Services provides fine granular controls of authorizing PKI administrators that are based on the certificate authority (CA) domain, the administrative action that is being performed, or the certificate type, as shown in Figure 3-3.



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
○ Show preregistered requests

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)
○ Show active, automatic renewal enabled certificates
○ Show active, automatic renewal disabled certificates
○ Show active, not renewable certificates

Additional search criteria (Optional)

Requestor's name

Show recent activity only (Not Selected)

Show certificates that will expire (Not Selected) (Only applicable to active certificates when recent activity is not selected)

Find Certificates or Certificate Requests

Home Page

email: webmaster@your-company.com

Figure 3-3 Administration web application sample window

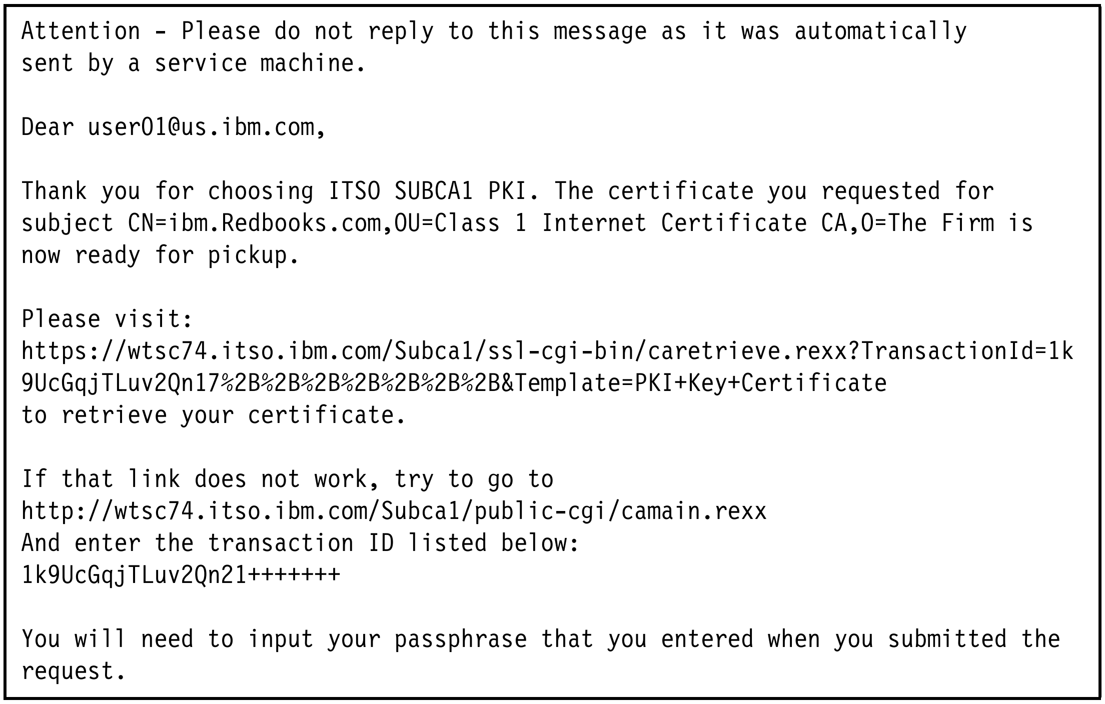
Depending on the IT Security policies you have in your enterprise, you might have a team of administrators in place for approving a certificate request. z/OS PKI Services provide the ability to require approvals from multiple administrators before issuing the requested certificate (NxM authorization).

In contrast to certificate requests that require multiple approval steps, you can set up z/OS PKI services to bypass the approval process; for example, the certificate request is automatically approved if the requester authenticated themselves in RACF first.

3.2.4 Email notifications

z/OS PKI Services provides the capability to send email notifications to different users in the following circumstances:

- Notify users whose certificate request was rejected or is ready for retrieval, as shown in Figure 3-4.
- Send renewal notifications to avoid issues arising from expired certificates. This ability helps prevent outages from occurring because of expired certificates, especially certificates owned by servers.
- Send email notifications to administrators who have requests pending.
- Automatically renew certificates.



```
Attention - Please do not reply to this message as it was automatically
sent by a service machine.

Dear user01@us.ibm.com,

Thank you for choosing ITSO SUBCA1 PKI. The certificate you requested for
subject CN=ibm.Redbooks.com,OU=Class 1 Internet Certificate CA,O=The Firm is
now ready for pickup.

Please visit:
https://wtsc74.itso.ibm.com/Subca1/ssl-cgi-bin/caretrieve.rexx?TransactionId=1k
9UcGqjTLuv2Qn17%2B%2B%2B%2B%2B%2B%2B&Template=PKI+Key+Certificate
to retrieve your certificate.

If that link does not work, try to go to
http://wtsc74.itso.ibm.com/Subca1/public-cgi/camain.rexx
And enter the transaction ID listed below:
1k9UcGqjTLuv2Qn21+++++++

You will need to input your passphrase that you entered when you submitted the
request.
```

Figure 3-4 Email notification that a requested certificate is available to pick up

3.2.5 Generating certificates

z/OS PKI Services acts as a CA and a Registration Authority (RA). It performs the verification, approval, and issuance processes.

If the certificate requester generated the key pair, the public key is sent with the request. The requester keeps their private key. z/OS PKI Services has no knowledge of the private key, but includes the public key into the certificate.

There is a key issue here that must be considered. Because z/OS PKI Services has no knowledge of the private key, z/OS PKI Services cannot recover the private key if the requester loses the private key. Also, the requester must recover the key or request a new certificate by using a new key pair.

However, if the requester asked z/OS PKI Services to generate the key pair (via ICSF), the key pair is stored it in the token data set. After the certificate is created, it is packaged with the private key and an email is sent to the requester that contains a link for the requester to retrieve the package.

z/OS PKI services can generate Rivest-Shamir-Adleman (RSA) keys and ECC keys.

3.2.6 Distributing certificates

Requesters can obtain their certificates by using the user web application directly from the web browser.

z/OS PKI Services also supports standardized protocols; therefore, client applications can request and obtain certificates autonomously. The support includes the following protocols:

- Simple Certificate Enrollment Protocol (SCEP)

  By using SCEP, you can securely issue certificates to large numbers of network devices by using an automatic enrollment technique. The network devices (often IPSEC devices, such as Cisco routers) must be SCEP-enabled and preregistered to your CA domain before they can successfully request certificates from you.

  You can configure PKI Services to respond automatically to SCEP certificate requests or to submit SCEP certificate requests to the PKI administrator for approval.

- Certificate Management Protocol (CMP)

  CMP is an Internet Protocol that is used to manage digital certificates within a PKI. A certificate request message object is used within the protocol to convey a request for a certificate to a CA. z/OS PKI Services allows a CMP client to communicate with it to request, revoke, suspend, and resume certificates.

3.2.7 Providing certificate revocation status

A certificate can be revoked by the administrator or the certificate owner. When a certificate is revoked, it is put on a Certificate Revocation List (CRL). If there are high activity levels for the certificates, the CRL can become large, which can result in longer response times for applications processing the CRL. If so, you might consider dividing the single CRL into Distribution Point CRLs (DP CRLs), as shown in Figure 3-5 on page 33.

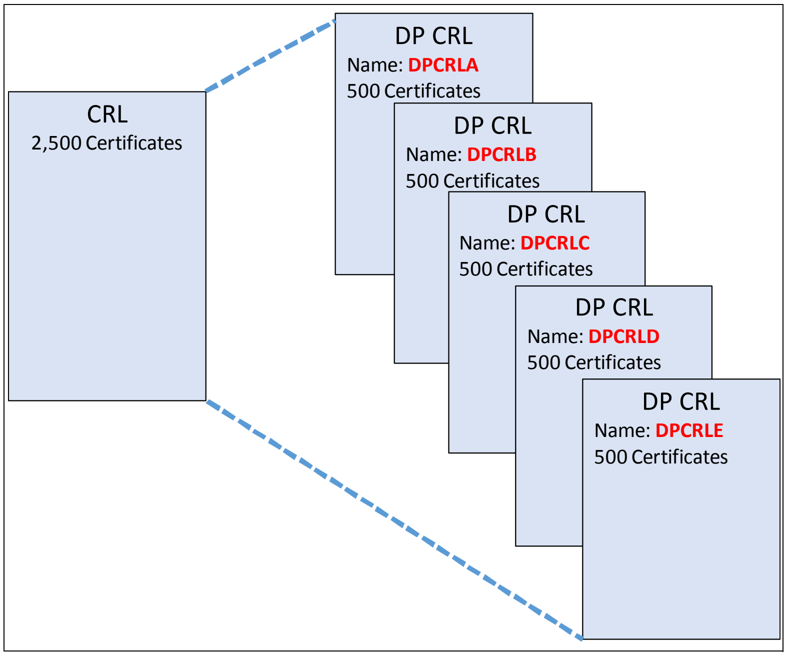


Figure 3-5 Dividing the CRL into multiple DP CRLs

Each DP CRL contains its own location and this location is contained within the `CRLDistributionPoints` extension value. The CRLs and the DP CRLs can be saved in a file system or they can be posted to LDAP. You check the revocation status by referring to the CRL or DP CRLs.

z/OS PKI Services provides you with another way to check the certificate revocations status by using the Online Certificate Status Protocol (OCSP). z/OS PKI Services can be enabled as an OCSP responder. Therefore, z/OS PKI Services can receive requests that contain a certificate serial number and it makes a call to OCSP, which used the certificate unique information to check the revocation status of the certificate and responds accordingly to the requester with the appropriate status. The location of the revocation status is contained within the certificate in the `AuthorityAccessInformation` extension.

Note: The status that is retrieved by using OCSP is current, although the CRL option might not reflect the latest content at the time the check was performed.

3.3 z/OS PKI Services elements overview

The elements that make up z/OS PKI Services can be divided into the following topics:

- User interfaces
- Request handlers
- Repositories
- Audit data and reporting opportunities

3.3.1 User interfaces

The interfaces can be started by a user or from a software on a server. The following interfaces are available:

- Certificate Management Protocol (CMP)
- Online Certitude Status Protocol (OCSP)
- Simple Certificate Enrollment Protocol (SCEP)
- Browser interface for the user
- Browser interface for the administrator

The interfaces are through the web services provided by the following servers:

- z/OS HTTP Server
- IBM WebSphere® Application Server

Figure 3-6 shows which interfaces can connect to which server.

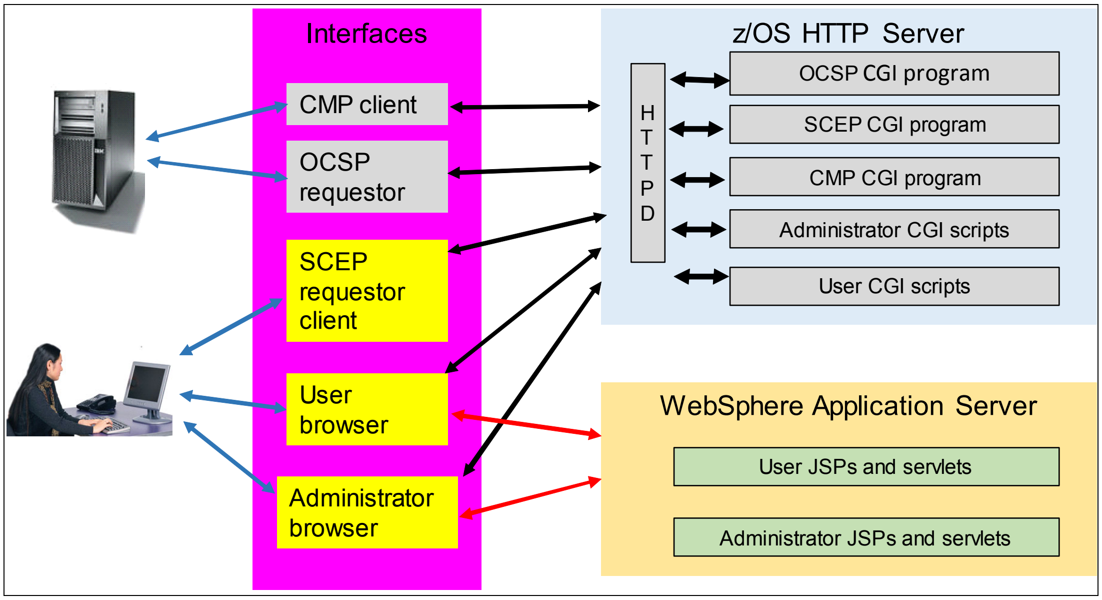


Figure 3-6 Interfaces

Note: IBM HTTP Server powered by Domino® V5.3 is not supported by z/OS V2R2. Earlier versions of z/OS PKI Services (z/OS 2.1 and lower) support IBM HTTP Server that is powered by Domino. The z/OS V2R2 requires the IBM HTTP Server powered by Apache V9.0, which is shipped with the base z/OS V2R2.

Certificate Management Protocol

CMP is an Internet Protocol that is used to manage X.509 digital certificates within a PKI. Messages can be sent to the CA to request, revoke, suspend, or resume a certificate. The CMP client sends the request directly to the HTTP server (and port number) that handles the client authentication requests. The CMP client must have a certificate installed in RACF (or equivalent) under the client's ID. The certificate is used by the requester to authenticate itself, and its owner ID is used to access the PKI.

Online Certificate Status Protocol

OCSP is used extensively for determining a certificate's status to see whether it was revoked. For more information, see "Providing certificate revocation status" on page 32.

Simple Certificate Enrollment Protocol

The web application can be used to pre-register as a client to use SCEP. The SCEP provides a simplified way for the z/OS PKI services web application to format the requests for the creation of certificate requests.

Browser interface for users

The user web pages consist of sample screens that you can easily customize to meet your organization's needs for certificate content and standards for appearance. For example, you can display your organization’s logo. It offers several certificate templates that the requester can use to create requests for various certificate types, based on the certificate’s intended purpose and validity period, and supports certificate requests that are automatically approved.

It supports the following tasks:

- Request a certificate
- Renew a certificate
- Revoke or suspend a certificate
- Pick up a certificate

Browser interface for administrators

The web application assists authorized administrators to manage certificate requests and issued certificates through their own web browsers. It also supports the following tasks:

- Reviewing pending certificate requests
- Querying pending requests to approve or reject those requests
- Displaying detailed information about a certificate or request
- Monitoring certificate information, such as validity period
- Annotating the reason for an administrative action
- Revoke or suspend a certificate
- Delete a certificate request
- Delete an issued certificate

Administrator and user actions

We described the z/OS PKI services functions that are available to the administrator and user. Table 3-1 lists the actions, required status of the certificate to take the action, and who can perform the action.

Table 3-1 Administrator and user actions summary

Action	Required Certificate Status	Who performs the action
Renew	Active	User
Resume	Suspended	Administrator
Revoke	Active or suspended	User or Administrator
Suspend	Active	User or Administrator
Delete	Active, Expired, Suspended, Revoked, or Revoked Expired	Administrator
Enable automatic renewal	Active or Active AutoRenewDisabled	Administrator
Disable automatic renewal	Active or Active AutoRenew	Administrator

Action	Required Certificate Status	Who performs the action
Change requester email	All statuses (applies only to certificates z/OS PKI Services generated the key pair	Administrator

3.3.2 Query and request handlers

The queries and requests from the user interfaces can be handled by the processes that are shown in Figure 3-7, depending on what actions are required to satisfy the request.

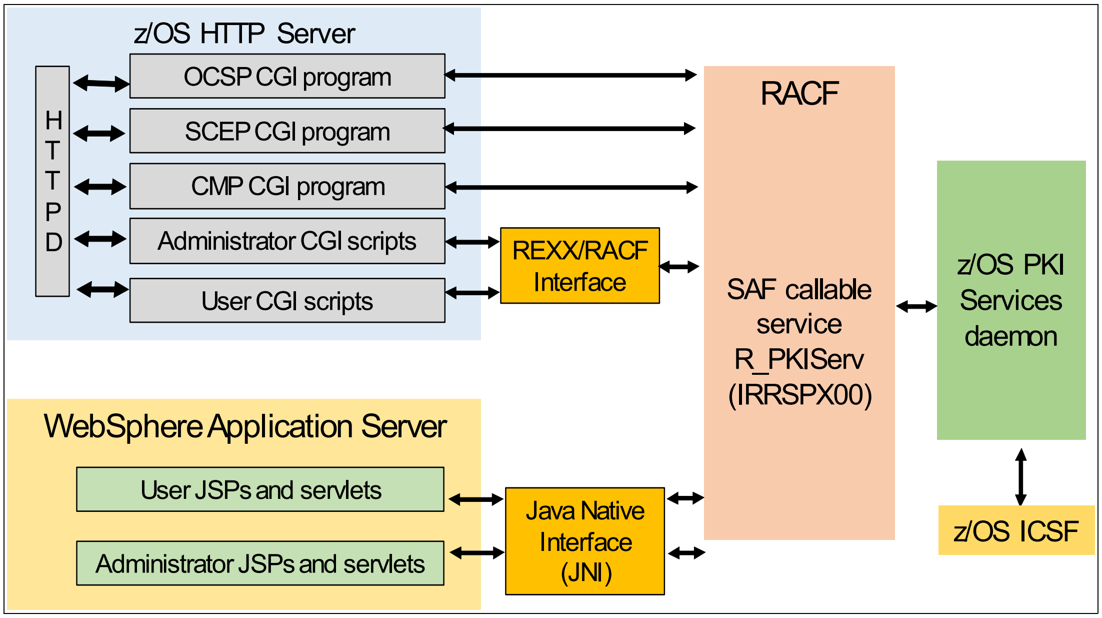


Figure 3-7 Query and request handlers

The queries and requests are captured by the HTTP server or WebSphere Application Server or go directly to the Systems Authorization Facility (SAF) Services application programming interface (API) R_PKIServ callable service (IRRSPX00). It allows authorized applications, such as servers, to programmatically request the functions of PKI Services. If the HTTP server is used, the CGI scripts call R_PKIServ through a REXX glue routine. If the WebSphere Application Server is used, the JSPs call R_PKIServ through the JNI layer.

RACF (or equivalent)

Controls can use the functions of the R_PKIServ callable service and protect the components of your PKI Services system. RACF creates your certificate authority's certificate, key ring, and private key. You can also use it to store the private key if ICSF is not available.

ICSF

Although optional, it is strongly suggested that ICSF is part of your PKI. Because it is not required to be present initially, it can be added later. PKI is necessary for the following functions:

- RACF can optionally use ICSF's public key data set (PKDS) to securely store the PKI Services CA signing key.
- PKI Services can use ICSF PKCS #11 token data set (TKDS) to store key pairs that PKI Services generates for non-CMP certificate requests.

> **Note:** If ICSF is not running and the TKSDS is not set up, PKI Services cannot generate the key pairs.

- Securely store the z/OS PKI Services certificate authority's private signing key and key pairs that PKI Services generates for certificates.

ICSF supports elliptic curve cryptography (ECC) keys. The z/OS PKI Services CMP CGI (see Figure 3-6 on page 34) needs ICSF's PKCS #11 to generate key pairs.

PKI Services daemon

The server daemon performs all the z/OS PKI Services functions.

3.3.3 Repositories

Figure 3-8 shows the repositories that are associated with z/OS PKI Services. Some of the repositories are optional and their existence and use depends on how you choose to configure your environment. For example, if you choose the IBM DB2® database options in preference to the VSAM options, DB2 must be available.

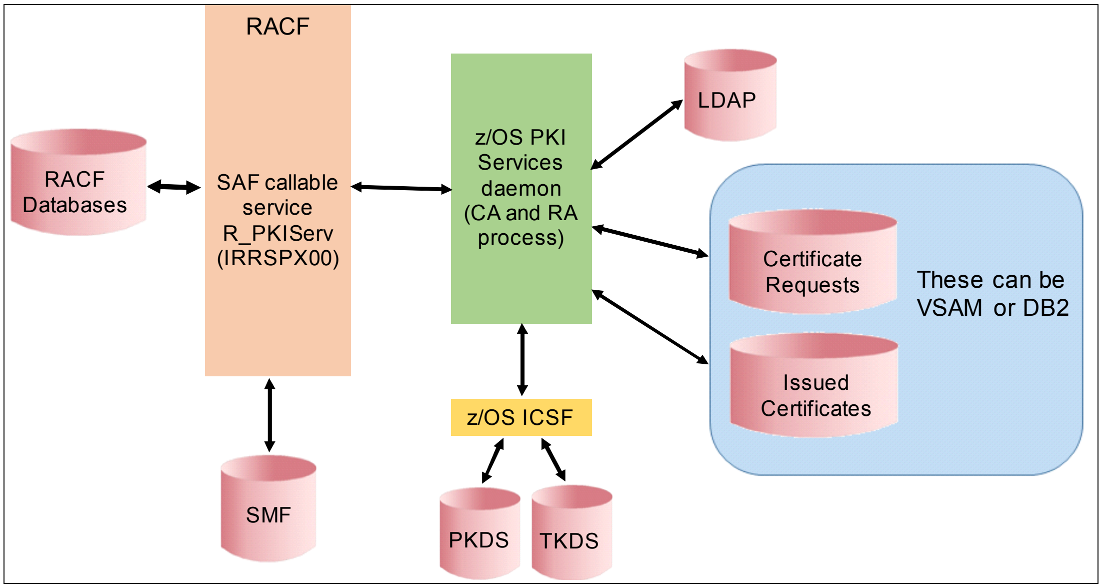


Figure 3-8 z/OS PKI Services associated repositories

The ICSF data sets also are optional. If you are not running ICSF, you cannot perform only specific tasks that are related to ICSF. You might not need those specific requirements.

RACF database

The RACF database contains many aspects that are related to security as a whole. In terms of z/OS PKI Services, it contains the following components:

- The CA's key ring, which contains the CA certificate and its private key
- Profiles protecting the PKI Services' functions and keys

LDAP

The directory that maintains information about the valid and revoked certificates that PKI Services issues in an LDAP-compliant format. You can use an LDAP server, such as the one provided by IBM Tivoli® Directory Server for z/OS.

Object store database

The object store contains all of the certificate requests. This store can be a VSAM file or DB2 database.

Issued certificates list database

This database contains the issued certificates list (ICL) and details about the certificates. This database can be a VSAM file or DB2 database.

Public key data set (optional)

The PKDS is used by RACF through ICSF to securely store the PKI Services CA signing key.

Token key data set (optional)

z/OS PKI Services can use ICSF PKCS #11 token data set (TKDS) to store key pairs that PKI Services generates.

3.3.4 Audit data and reporting opportunities

z/OS PKI services creates SMF type 80 records through the `RACF R_PKIServ` callable service and through the daemon. Figure 3-9 shows a list of the recorded activities.

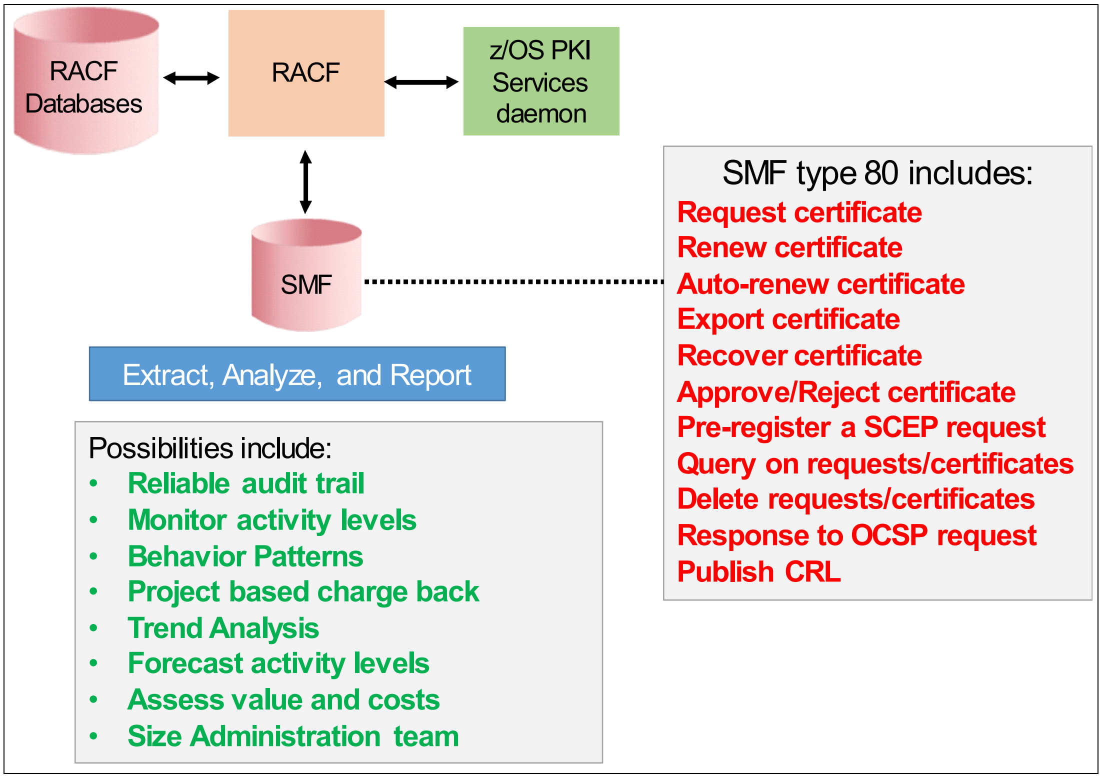


Figure 3-9 SMF data capture and possible uses

The SMF record type 80 data can be post-processed in similar ways as other SMF records are used for valuable reporting on activity levels.

Audit trail

The SMF data is an audit trail of the z/OS PKI Services activities. Depending on the audit policies and practices that are deployed in your site, this feature is an effective way to report certificate activities to monitor the policies and practices and to measure their effectiveness. The SMF data can be analyzed and includes the following possibilities for its use:

- To identify non-expired obsolete certificates from remaining available
- Monitor the request activities and identify who is involved with each request
- Provide activity level for potential charge back to projects
- Analyze activity trends
- Explore certificate status for problem determination purposes

3.4 Added value of z/OS PKI Services

The z/OS PKI Services is a full PKI and can support the certificate lifecycle. It also adds value that is inherited from the underlying hardware and operating system.

3.4.1 Scalability

The implementation of your PKI infrastructure can enjoy the scalability of z Systems hardware and software. You can use the amount of capacity you need to fulfil your organization’s requirements.

3.4.2 Availability

z/OS PKI Services benefits from the Qualities of Service for availability. This benefit is provided by the underlying operating System z/OS, especially when z/OS Sysplex capabilities are used.

The scope of sharing affects your strength of availability. With the Sysplex option offered by z Systems, you can share the repositories across the sysplex; that is, across multiple instances of z/OS. Availability is strengthened because each z/OS system can access z/OS PKI Services and access the shared data.

Figure 3-10 on page 40 shows a scenario in which there is a remote data center that also has a Sysplex. This configuration can take over the certificate work if the first data center become unavailable.

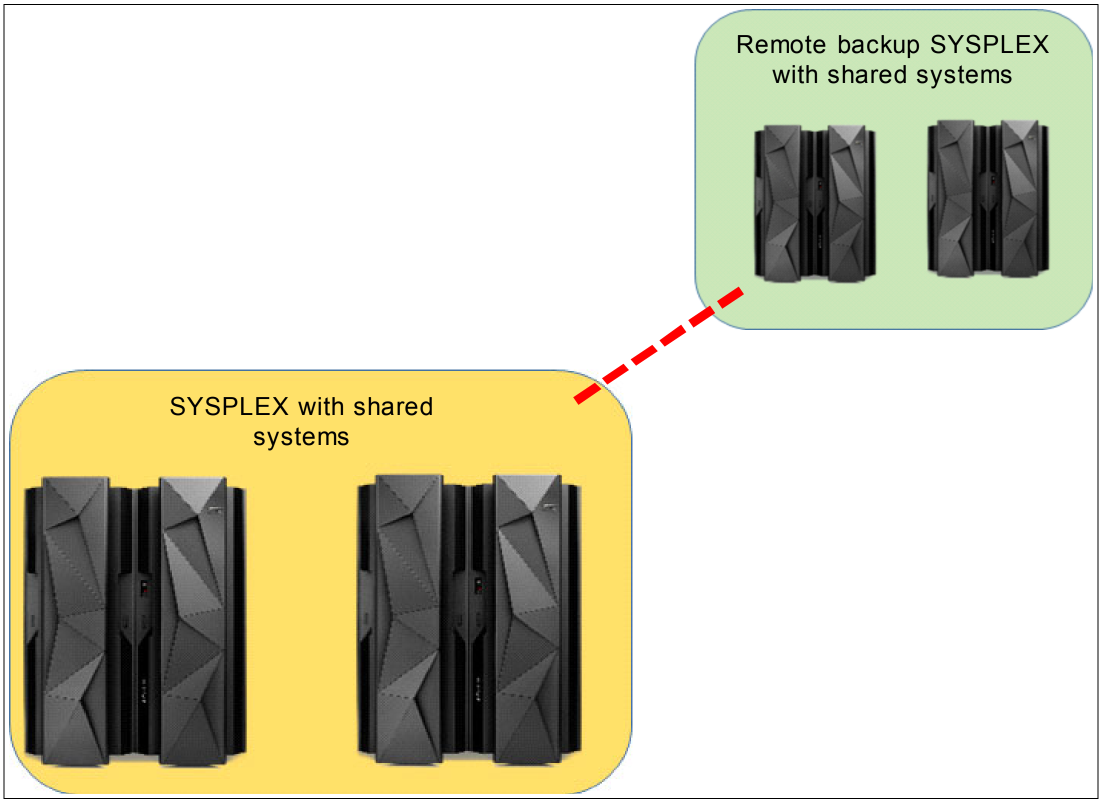

Remote backup SYSPLEX with shared systems
SYSPLEX with shared systems

Figure 3-10 Availability

When VSAM data sets are used as back-end storage, VSAM record level sharing (RLS) can be used across the Sysplex. When DB2 for z/OS is used, DB2 data sharing can be used to set up high available back-end storage. IBM HTTP Server and WebSphere Application Server allow setup for failover scenarios.

3.4.3 Security

PKI Services uses the SSL/TLS for all the traffic flow through the z/OS HTTP server. The WebSphere Application Server can be used as an alternative to the z/OS HTTP Server, which is the same security mechanisms regarding encrypted traffic and authenticated requests apply.

The Resource Access Control Facility (RACF) or equivalent external security manager can be used to control who can call the PKI Services functions and protect access of the private key of the CA.

RACF provides granular administration authorization control on requests and certificates based on the domain, action, and the certificate template.

RACF creates your CA's certificate, key pair, certificate, and key ring. The private key is stored in the RACF database or in the public key data set (PKDS) if ICSF is available.

If the Enterprise PKCS#11 coprocessor is available to your system, it provides the ability for hardware protection for the private key of the PKI Services CA and those it generated for the requesters.

The private signing key of the CA is critical. If this key is compromised, an attacker can issue certificates in the name of the CA. The use of a hardware security module (HSM), such as the IBM Enterprise PKCS#11 coprocessor, ensures that the CA's private signing key never leaves the secure coprocessor boundary decrypted.

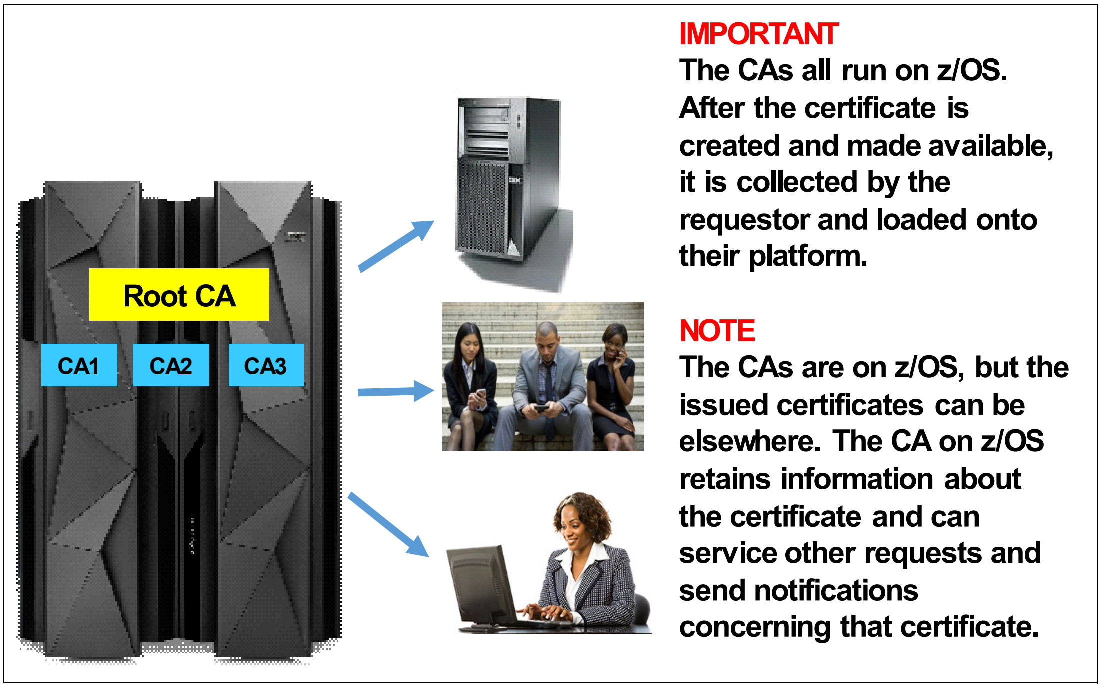
3.4.4 Cost

z/OS PKI Services is not a separately priced product. Rather, it is licensed and integrated within z/OS.

Because z/OS PKI Services provides the capability to issue certificates, it is an alternative to buying certificates from third parties; for example, for the company's internal IT environment. With costs of $20 - $1,200 USD per SSL certificate and year for x number of entities, this issue can easily sum up to millions of savings per year.

3.5 Certificates across the enterprise

Figure 3-11 shows an example for a possible CA hierarchy set up in an enterprise. The root CA has 3 intermediate or sub CAs.



IMPORTANT
The CAs all run on z/OS. After the certificate is created and made available, it is collected by the requestor and loaded onto their platform.

NOTE
The CAs are on z/OS, but the issued certificates can be elsewhere. The CA on z/OS retains information about the certificate and can service other requests and send notifications concerning that certificate.

Figure 3-11 Possible setup of enterprise CA hierarchy

The three sub CAs issue certificates for different use cases in the enterprise. You can divide the sub CAs into organizational structures to match your enterprise.

Different use case can be for example, certificates to authenticate mobile devices to the corporate network, SSL certificates for internal servers, VPN certificates for users dialing into the corporate network from home, and certificates to sign or encrypt email.

After the certificate is available, the requesters can download it to their respective areas. This task can be on any platform or in the requester's cloud.

3.6 What is next?

You now have a high-level understanding of the capabilities of z/OS PKI services and the key role it can play in providing an effective way to create and manage certificates and provide you with important email notification functions and audit trails to help reduce security risks.

The next step for you is to implement a quick and simple structure on a test LPAR so that you can quickly see and get the feel of z/OS PKI services.

This book shows you how to set up and use the web application with the default supplied templates.

For more information, see the IBM Redbooks publication *IBM PKI on z/OS: Quick Set up and Explore*, SG24-8337, which is planned for publication in March 2016.

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topics in this document. Some publications that are referenced in this list might be available in softcopy only:

- *IBM z/OS V1R12 Communications Server TCP/IP Implementation: Volume 4 Security and Policy-Based Networking*, SG24-7899
- *IBM z/OS V2R2: Security*, SG24-8288

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft, and other materials, at the following website:

ibm.com/redbooks

Other publications

The following publications are also relevant as further information sources:

- *z/OS Security Server RACF Security Administrator's Guide*, SA23-2289
- *z/OS Security Server RACF Security System Programmer's Guide*, SA23-2287
- *IBM Encryption Facility for z/OS: Planning and Customizing*, SA23-2229

Online resources

These websites are also relevant as further information sources:

- Internet x.509 PKI and CRL Profile:

  https://www.ietf.org/rfc/rfc5280.txt
- Article: *Drowning in digital certificates? Here's a lifeline!*:

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

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

SG24-8336-00

ISBN 0738441503

Printed in U.S.A.