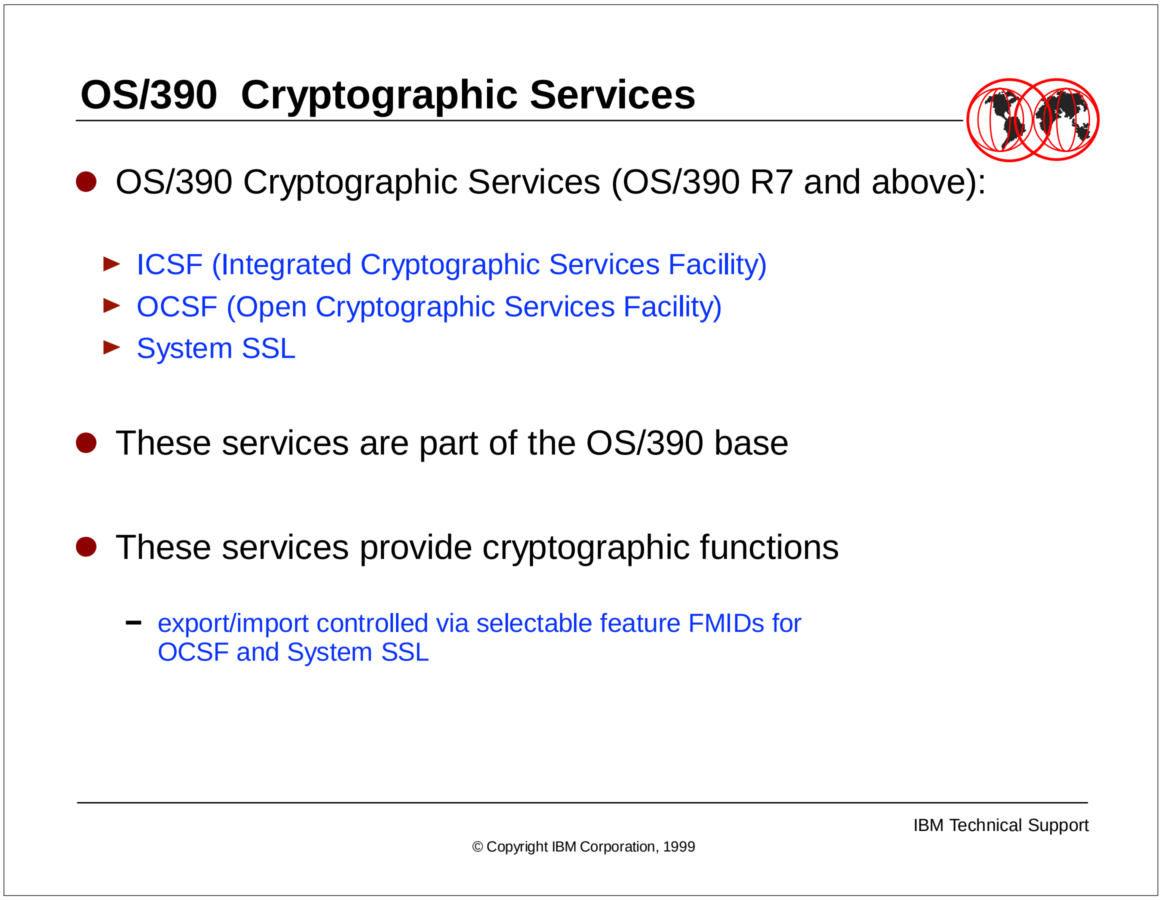
Common Data Security Architecture on OS/390

Paul de Graaff ITSO Poughkeepsie S/390 Security



OS/390 Cryptographic Services

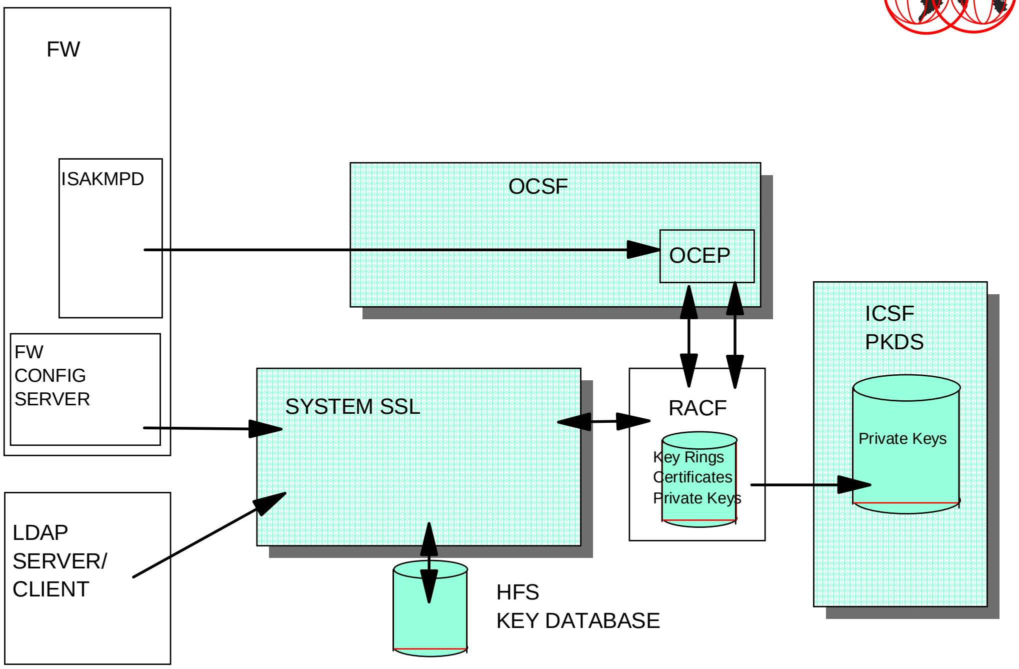
- OS/390 Cryptographic Services (OS/390 R7 and above):
  - ► ICSF (Integrated Cryptographic Services Facility)
  - ► OCSF (Open Cryptographic Services Facility)
  - ► System SSL
- These services are part of the OS/390 base
- These services provide cryptographic functions
  - export/import controlled via selectable feature FMIDs for OCSF and System SSL

IBM Technical Support

© Copyright IBM Corporation, 1999

- ▸ A new base element has been defined for OS/390 R7 and above: the OS/390 Cryptographic Services.
- ▸ OS/390 Cryptographic Services is an exclusive, that is not separately orderable, element of OS/390.
- ▸ OS/390 Cryptographic Services comprise ICSF, OCSF (new at OS/390 R7) and System SSL (new at OS/390 R7).
- ▸ There are several associated features to control the exportation/importation of cryptographic services in OCSF and System SSL. The ICSF algorithms exportation control is achieved by getting the proper hardware enablement diskette for the cryptographic coprocessor.

OS/390 Cryptographic Services in OS/390 R8



IBM Technical Support

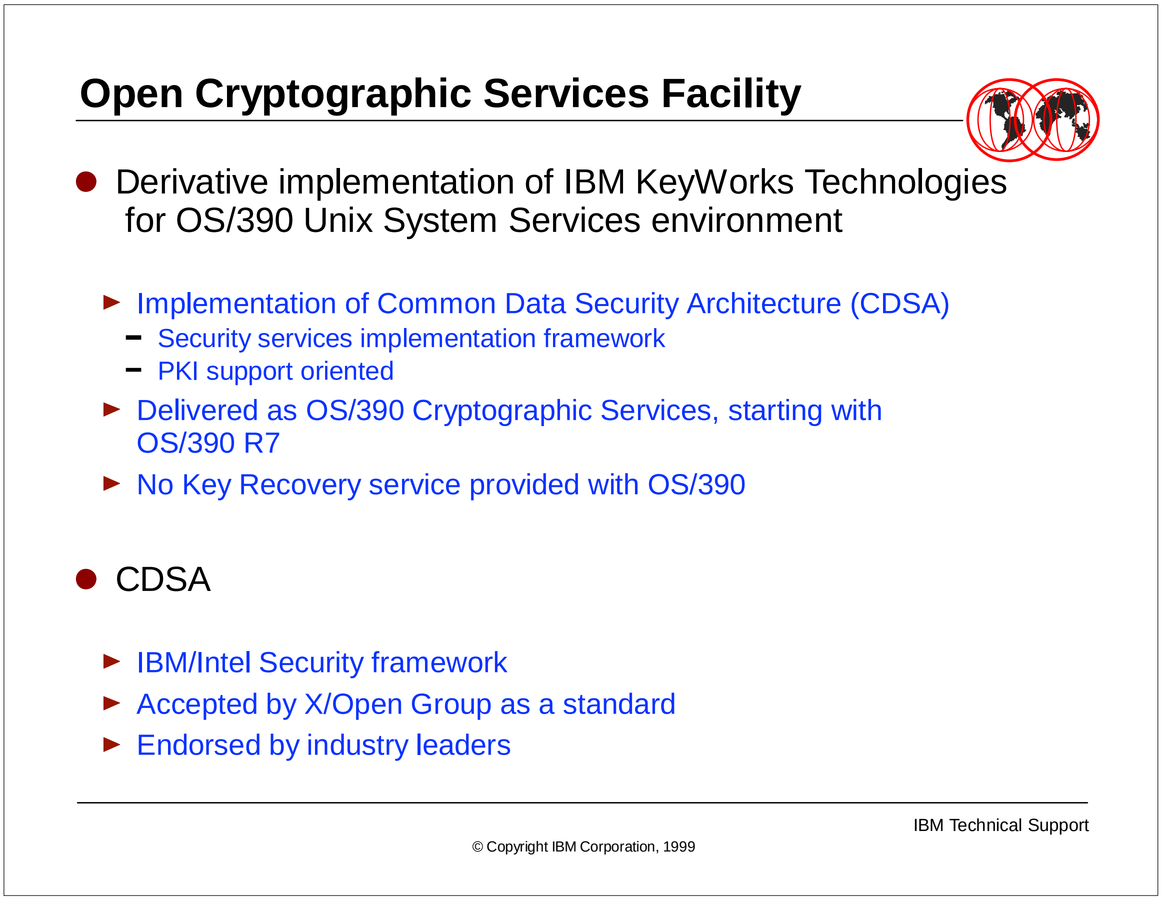
© Copyright IBM Corporation, 1999

- The following interactions exist between the elements of cryptographic services in OS/390, as of OS/390 R8:
- OCSF is used by the ISAKMPD daemon of the OS/390 Firewall Technologies. The ISAKMPD daemon allows the two ends of an IPSec tunnel to dynamically negotiate security associations and session keys.More precisely, ISAKMPD is using the OCEP plug-ins for OCSF to access keys and certificates managed by RACF.
- The Firewall Technologies configuration server, that is the daemon in the firewall host that communicates with the configuration GUI workstation, is using system SSL to secure the conversation with the workstation. System SSL can use keys and certificates stored in HFS files (key databases) or stored in the RACF Database. Note that System SSL does not support, as of today, using private keys stored in the ICSF PKDS.
- RACF can keep private keys either in the RACF Database or in the ICSF PKDS (Public Key Data Set). The latter is obtained with the option ICSF of the RACDCERT ADD or RACDCERT GENCERT commands.
- The OS/390 LDAP server and LDAP client are also using System SSL to secure their communications.

Open Cryptographic Services Facility

IBM Technical Support

© Copyright IBM Corporation, 1999



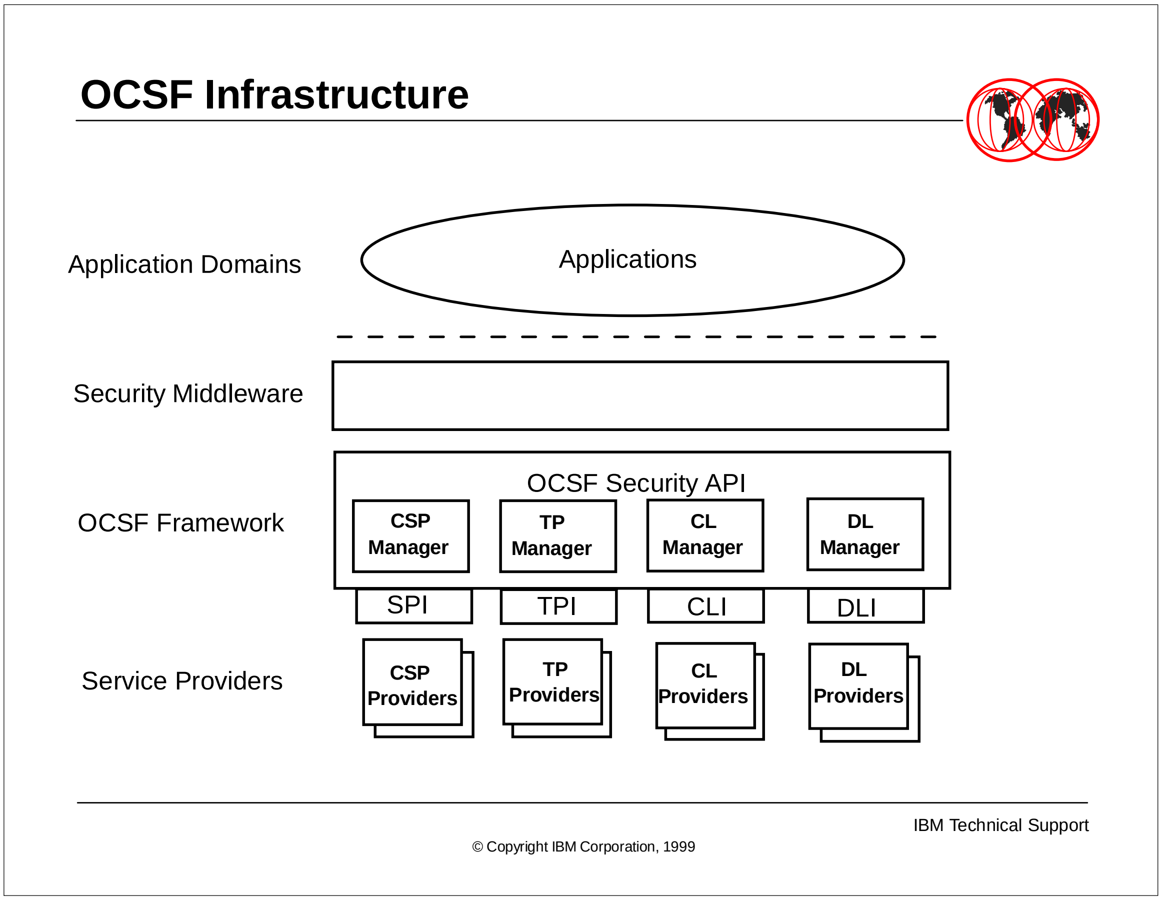
Open Cryptographic Services Facility

- Derivative implementation of IBM KeyWorks Technologies for OS/390 Unix System Services environment
  - Implementation of Common Data Security Architecture (CDSA)
    - Security services implementation framework
    - PKI support oriented
  - Delivered as OS/390 Cryptographic Services, starting with OS/390 R7
  - No Key Recovery service provided with OS/390
- CDSA
  - IBM/Intel Security framework
  - Accepted by X/Open Group as a standard
  - Endorsed by industry leaders

IBM Technical Support

© Copyright IBM Corporation, 1999

- Intel has developped the CDSA (Common Data Security Architecture) framework to support, via a single API, multiple security related services, such as encryption, certificate management, key recovery, trust policy and data storage. These services are strongly oriented towards PKI support.
- The CDSA security services are implemented in plug-in modules called "service providers".
- CDSA is supported by the IBM Secure Cryptography and Certificate Services (SCCS) Toolkit, part of the IBM KeyWorks family of products. It consists of a framework (SCCS Framework) and several service provider add-in modules.
- The SCCS Toolkit has been ported to OS/390 at OS/390 R7 and renamed "Open Cryptographic Services Facility", which is now a base element of OS/390. Note that there is no Key Recovery service provider in OCSF.
- CDSA has been adopted by The Open Group as an industry-accepted specification for the development of secure applications, which offers extensibility and cross-platform support.



OCSF Infrastructure

Application Domains

Applications

Security Middleware

OCSF Framework

OCSF Security API

CSP Manager

TP Manager

CL Manager

DL Manager

SPI

TPI

CLI

DLI

Service Providers

CSP Providers

TP Providers

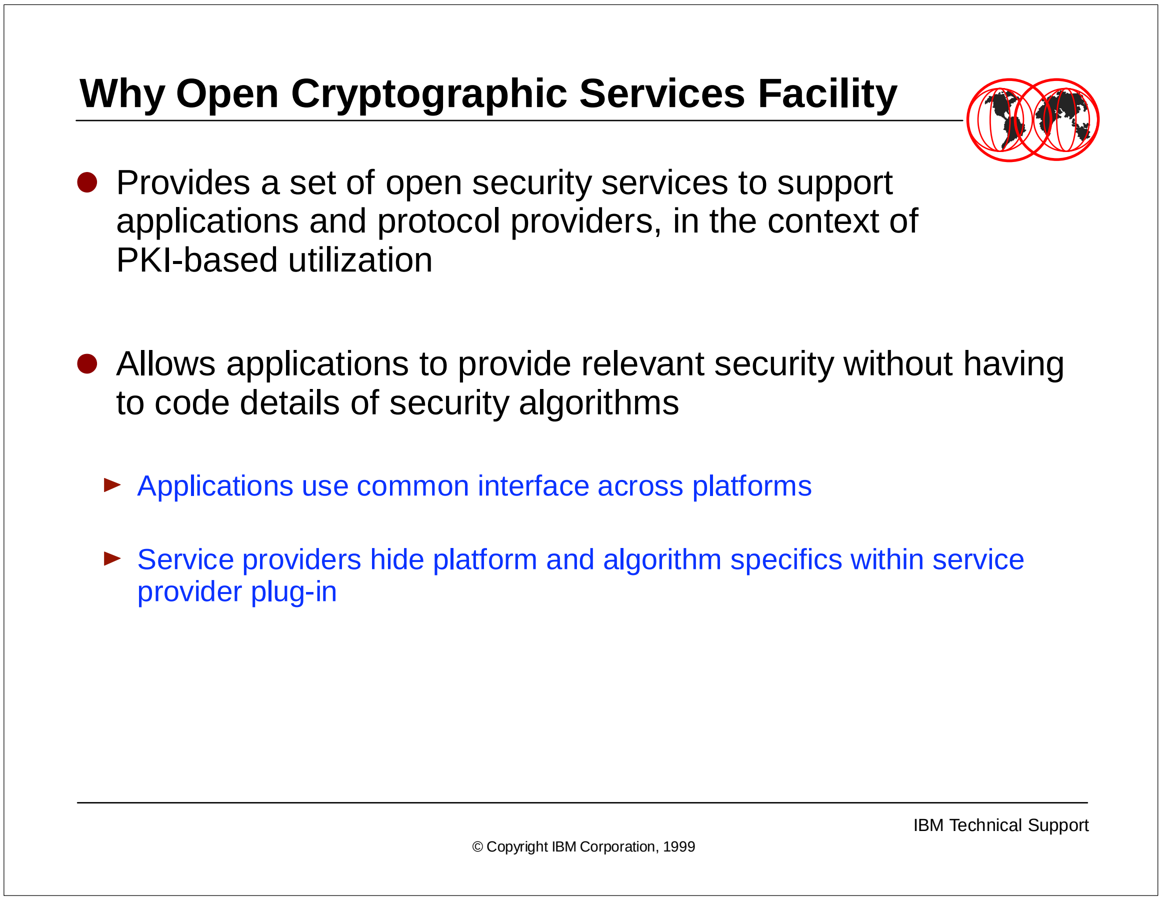
CL Providers

DL Providers

IBM Technical Support

© Copyright IBM Corporation, 1999

- OCSF is a set of layered security services and associated APIs.
- OCSF layers comprise
- At the "bottom": fundamental services, such as cryptographic algorithms, given by "service providers".
- In the "middle": the OCSF Framework layer, which acts as a switch between the applications or security middleware.
- At the "top", the applications or security middleware that require cross-platform uniform access to security services.
- The OCSF Framework provide this uniform access and acts as a switch logically connecting an application/middleware to the service provider giving the required service. The interface with the proper service provider category is managed by a service "Manager".
- There are four categories of services provided by OS/390 OCSF:
  - Cryptographic Services.
  - Trust Policy services.
  - Certificate Library service.
  - Data Library services.



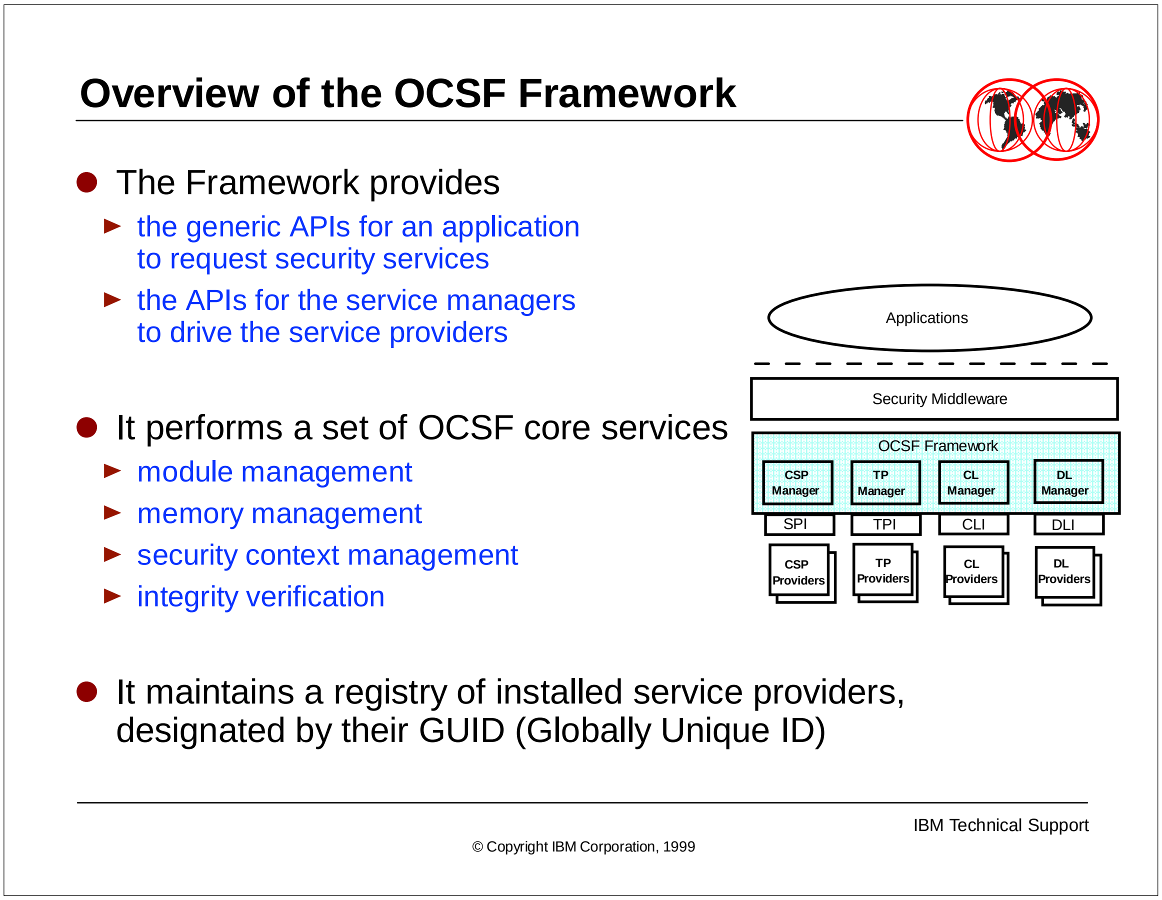
Why Open Cryptographic Services Facility

- Provides a set of open security services to support applications and protocol providers, in the context of PKI-based utilization
- Allows applications to provide relevant security without having to code details of security algorithms
  - Applications use common interface across platforms
  - Service providers hide platform and algorithm specifics within service provider plug-in

IBM Technical Support

© Copyright IBM Corporation, 1999

- The intent of OCSF is to answer the requirement of making cryptographic services as a base service in the operating system.
- The available services are intended to provide the functions required by PKIs (Public Key Infrastructure), that is are certificate-based.
- to have these services accessible using a high level API common across several platforms.
- to provide openess by allowing new services to fit within the OCSF Framework as service provider "plug-ins"



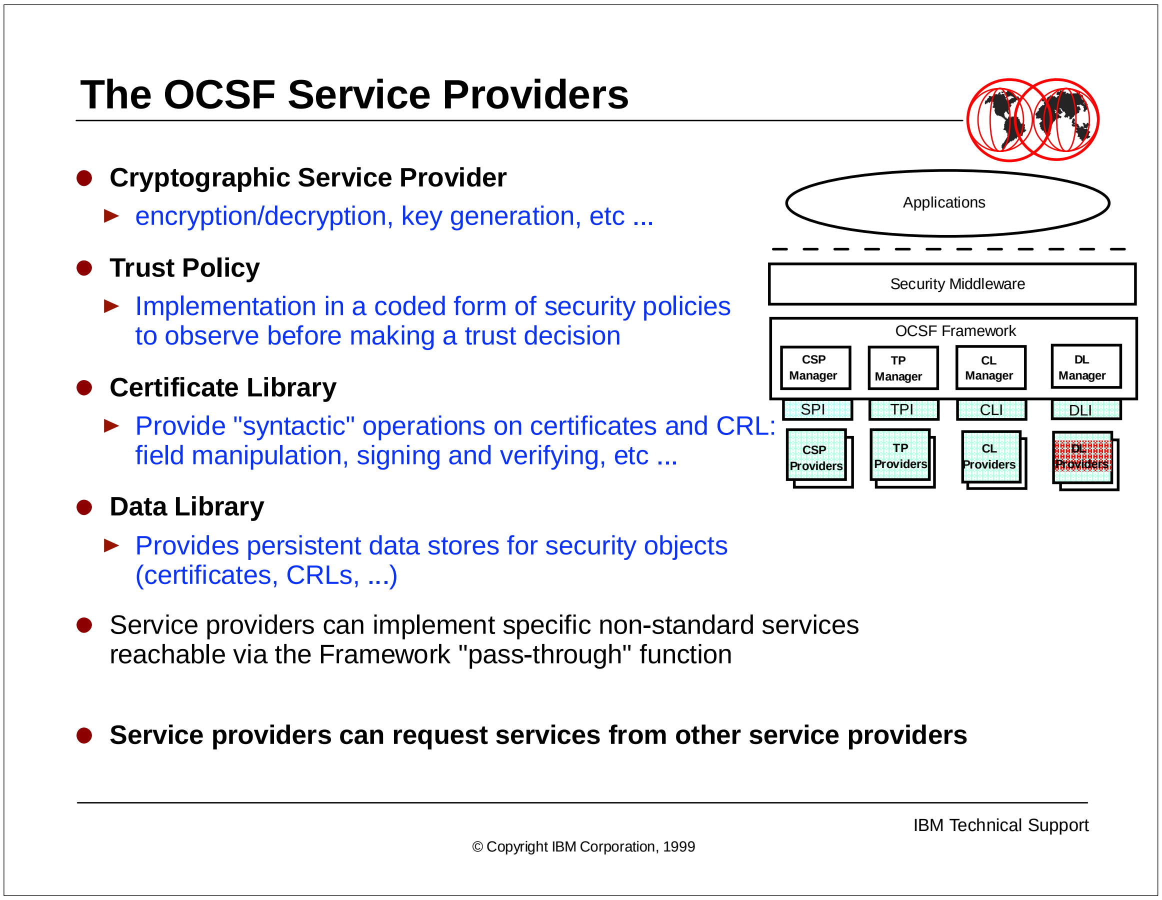
Overview of the OCSF Framework

- The Framework provides
  - the generic APIs for an application to request security services
  - the APIs for the service managers to drive the service providers
- It performs a set of OCSF core services
  - module management
  - memory management
  - security context management
  - integrity verification
- It maintains a registry of installed service providers, designated by their GUID (Globally Unique ID)

IBM Technical Support

© Copyright IBM Corporation, 1999

- The OCSF Framework is providing extensibility and adherence to open standards by providing two sets of APIs:
  - The APIs that an application uses to call for OCSF services.
  - The APIs that a service provider manufacturer must use to make the module pluggable within the OCSF Framework and manageable by the OCSF service managers.
- There are no limitations on the number of service providers that can be invoked in OCSF. Each service provider is registered at installation time in the OCSF Framework using a GUID (Globally Unique ID). This is actually part of the module management function described below.
- The framework will logically "attach" the service provider designated to fulfill the application's request.
- The Framework also provides core OCSF services:
  - Module management - installation/uninstallation, registration, dynamic loading of service provider modules.
  - Memory management - decision to free an application owned storage object.
  - Security context management - instantiation and management of user owned secured data structures ("security contexts").
  - Integrity verification - verification of service providers code integrity. The verification of code integrity is based on the use of the program control feature of RACF. OCSF requires all codes including OCSF binaries and the invoking



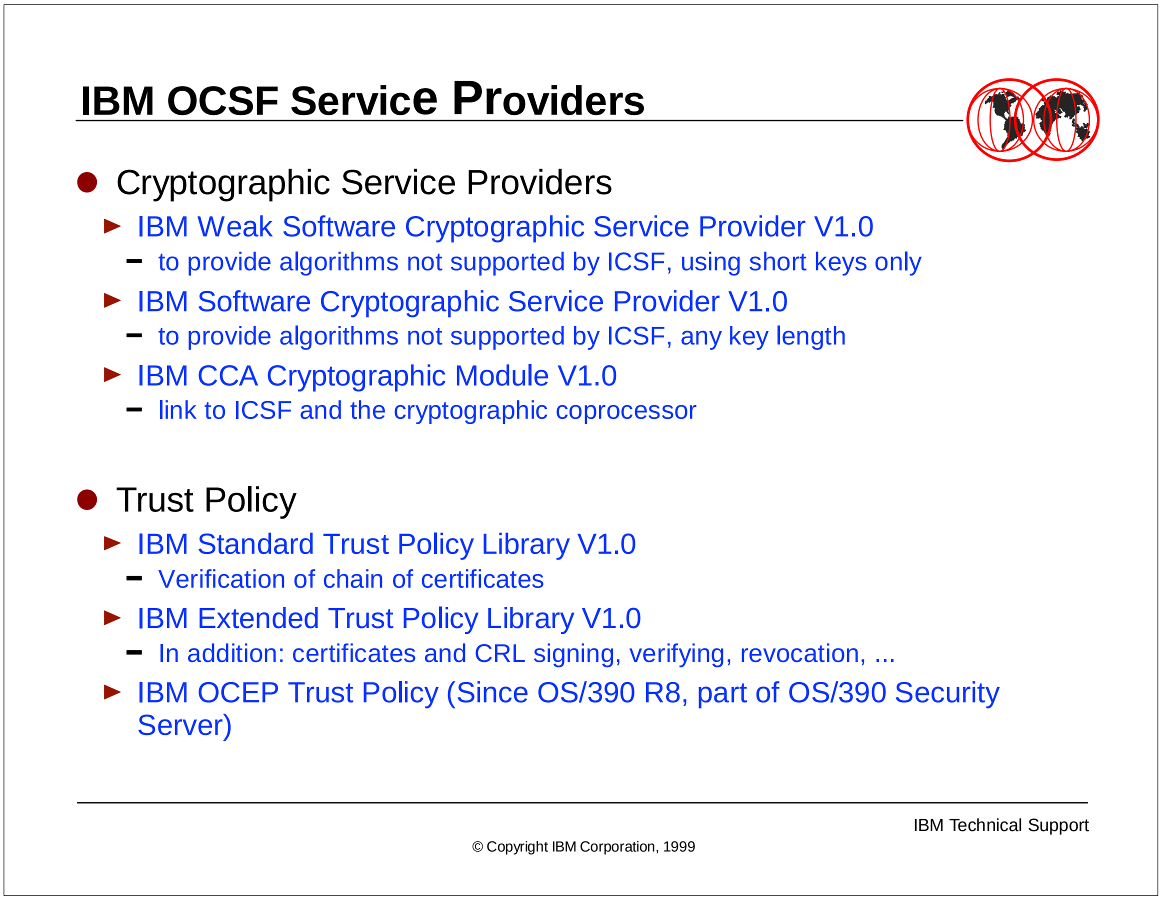
The OCSF Service Providers

- **Cryptographic Service Provider**
  - encryption/decryption, key generation, etc ...
- **Trust Policy**
  - Implementation in a coded form of security policies to observe before making a trust decision
- **Certificate Library**
  - Provide "syntactic" operations on certificates and CRL: field manipulation, signing and verifying, etc ...
- **Data Library**
  - Provides persistent data stores for security objects (certificates, CRLs, ...)
- Service providers can implement specific non-standard services reachable via the Framework "pass-through" function
- **Service providers can request services from other service providers**

IBM Technical Support

© Copyright IBM Corporation, 1999

- Service providers are plug-ins driven by the service managers.
- There are four categories of service providers in OCSF, that performs a specialized set of functions accessible through the related service manager interface (Trust Policy Interface, Certificate Library Interface, etc ...).
  - The Cryptographic Service Providers provide all cryptographic functions required by PKIs.
  - The Trust Policy service providers are implementing, in a coded form, the policies defined for a specific security by the installation or institution(s), to abide with before making a trust decision.
  - The Certificate Library provides high level syntactic functions on certificates and certificate revocation lists. The functions provided are strongly related to the format of certificates and CRLs.
  - The Data Library provides persistent storage for security objects. Where these objects can be stored into and retrieved from.
- service providers can implement specific services for which no API have been defined in OCSF. The OCSF Framework "pass-through" function call allows to directly call a specific non-standard service as per the documentation provided with the targetted service provider.
- It is usual that a service provider calls another service provider to perform the complete service it has to provide. As an example, a Trust Policy service provider may have to verify a certificate’s signature. The Trust Policy service provider will



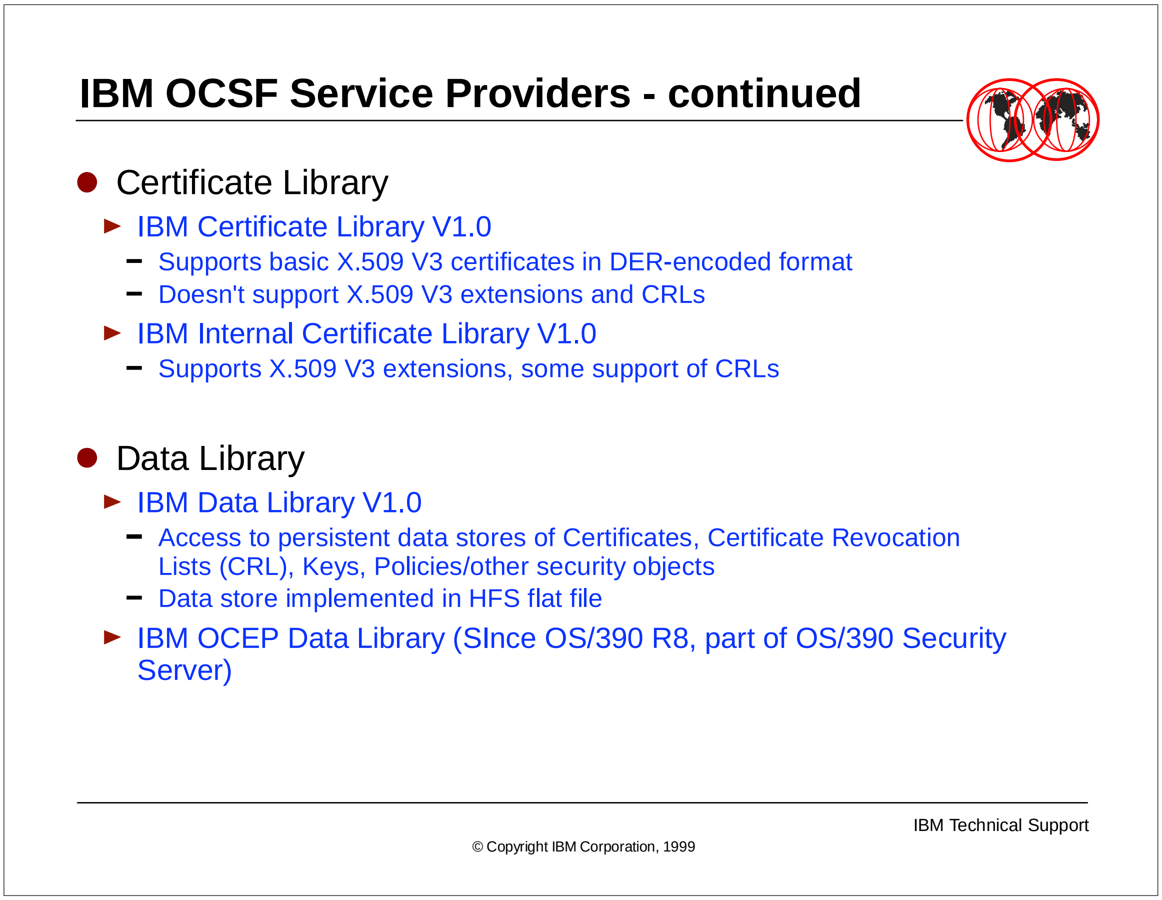
IBM OCSF Service Providers

- Cryptographic Service Providers
  - IBM Weak Software Cryptographic Service Provider V1.0
    - to provide algorithms not supported by ICSF, using short keys only
  - IBM Software Cryptographic Service Provider V1.0
    - to provide algorithms not supported by ICSF, any key length
  - IBM CCA Cryptographic Module V1.0
    - link to ICSF and the cryptographic coprocessor
- Trust Policy
  - IBM Standard Trust Policy Library V1.0
    - Verification of chain of certificates
  - IBM Extended Trust Policy Library V1.0
    - In addition: certificates and CRL signing, verifying, revocation, ...
  - IBM OCEP Trust Policy (Since OS/390 R8, part of OS/390 Security Server)

IBM Technical Support

© Copyright IBM Corporation, 1999

- All these services providers, except for the OCEP Trust Policy library, are delivered as part of OS/390 OCSF. However they come in the selectable security features and as such, must be installed.
- There are three Cryptographic Service Providers available :
  - two software CSPs, depending on the security feature only one (the weak software CSP is installed) or both the weak software CSP and the software CSP are installed.
  - Note: the OS/390 OCSF software CSPs are products licensed from RSA Data Security, Inc. Proper arrangement must be made with RSA whether the software CSP are to be used for development and test of applications, or wether they will be used within a marketted application. The OCSF Application Programmer's Guide and Reference, SC24-5875, provides the detailed procedure to follow.
  - A hardware, or CCA (IBM Common Cryptographic Architecture), CSP. This CSP interfaces with ICSF and the hardware cryptographic coprocessor. This CSP is not subject to selection by the OCSF security features by themselves, as the enablement of cryptographic algorithms inside the crypto coprocessor depends on the hardware configuration diskette shipped for the very coprocessor.
- There are two Trust Policy service providers delivered. The standard library offers onlv certificate chain verification. whereas the extended librarv provides manv



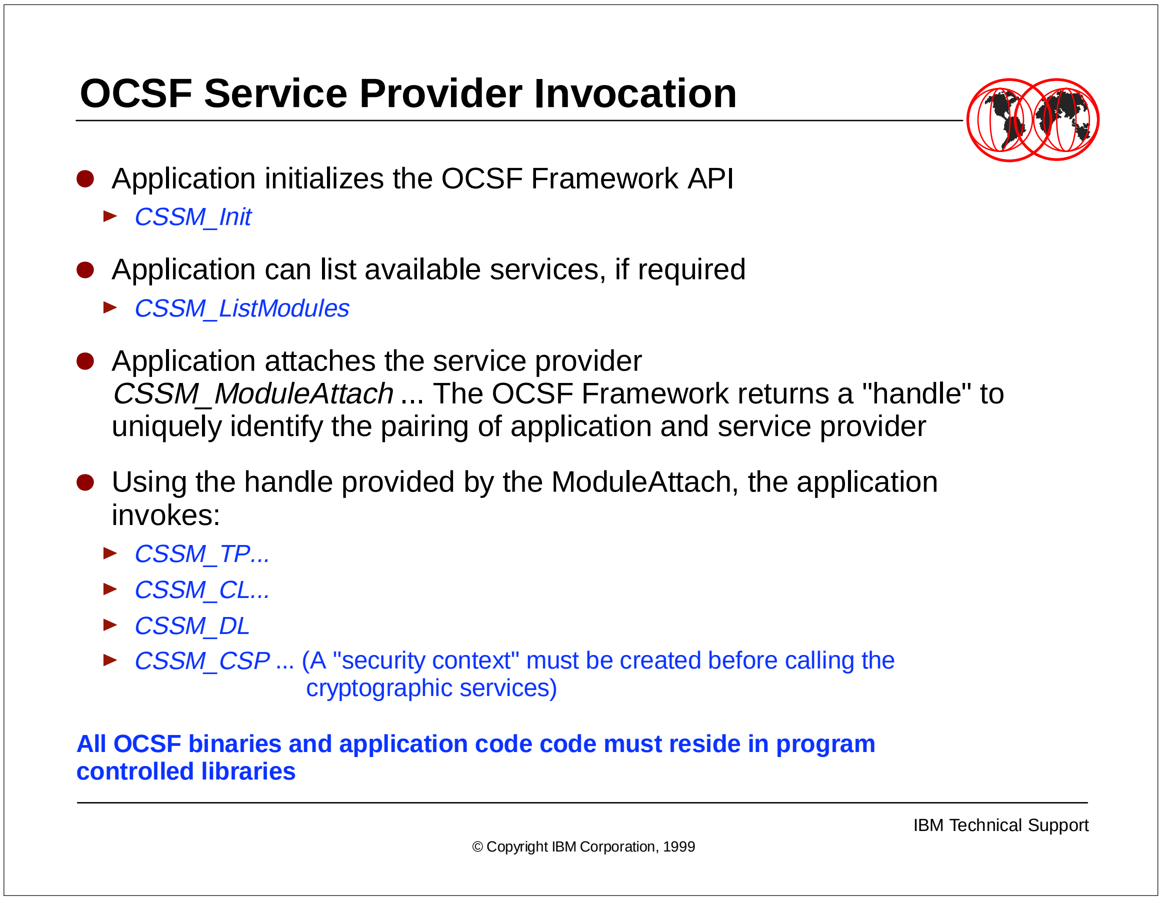
IBM OCSF Service Providers - continued

- Certificate Library
  - ► IBM Certificate Library V1.0
    - Supports basic X.509 V3 certificates in DER-encoded format
    - Doesn't support X.509 V3 extensions and CRLs
  - ► IBM Internal Certificate Library V1.0
    - Supports X.509 V3 extensions, some support of CRLs
- Data Library
  - ► IBM Data Library V1.0
    - Access to persistent data stores of Certificates, Certificate Revocation Lists (CRL), Keys, Policies/other security objects
    - Data store implemented in HFS flat file
  - ► IBM OCEP Data Library (SInce OS/390 R8, part of OS/390 Security Server)

IBM Technical Support

© Copyright IBM Corporation, 1999

- All these services providers, except for the OCEP Trust Policy library, are delivered as part of OS/390 OCSF. However they come in the selectable security features and as such, must be installed.
- There are two Certficate Libraries. The "internal" one is used by IBM products and not documented.
- The Certificate Library V1.0 supports functions such as:
  - signing and verifying of certificates.
  - extraction of certificate fields.
  - extraction of public key from certificates.
- There is one Data Library coming with OCSF, using an HFS file as the object repository.
- There is one Data Library coming with OCEP, as part of the OS/390 Security Server.



OCSF Service Provider Invocation

- Application initializes the OCSF Framework API
  - *CSSM_Init*
- Application can list available services, if required
  - *CSSM_ListModules*
- Application attaches the service provider *CSSM_ModuleAttach* ... The OCSF Framework returns a "handle" to uniquely identify the pairing of application and service provider
- Using the handle provided by the ModuleAttach, the application invokes:
  - *CSSM_TP...*
  - *CSSM_CL...*
  - *CSSM_DL*
  - *CSSM_CSP* ... (A "security context" must be created before calling the cryptographic services)

All OCSF binaries and application code code must reside in program controlled libraries

IBM Technical Support

© Copyright IBM Corporation, 1999

- The application uses CSSM_ * C/C++ functions to call for OCSF services.
- The application
  - instantiates and initializes the OCSF API.
  - locates the service provider module, if necessary, by calling the registry services of the OCSF Framework. It then gets the GUID of the module.
  - "Attaches" the desired module using its GUID. The OCSF Framework then returns a "handle" uniquely identifying the pairing of the application and the attached service provider.
  - Calls the attached service provider services, using the handle, with functions of the form CSSM_TP_... (for Trust Policy services), CSSM_CL_... (Certificate Library), CSSM_DL_... (Data Library), and CSSM_CSP_...(Cryptographic Service Provider).
- Note that prior to calling a cryptographic service provider, the application must create a "security context", that is a data structure used as input/output by the service provider. A "security context" is secured in that it is uniquely identified by a context "handle" passed to the requestor at context creation by the OCSF Framework.
- The OCSF code itself and the application code must not run in a "dirty" address space.

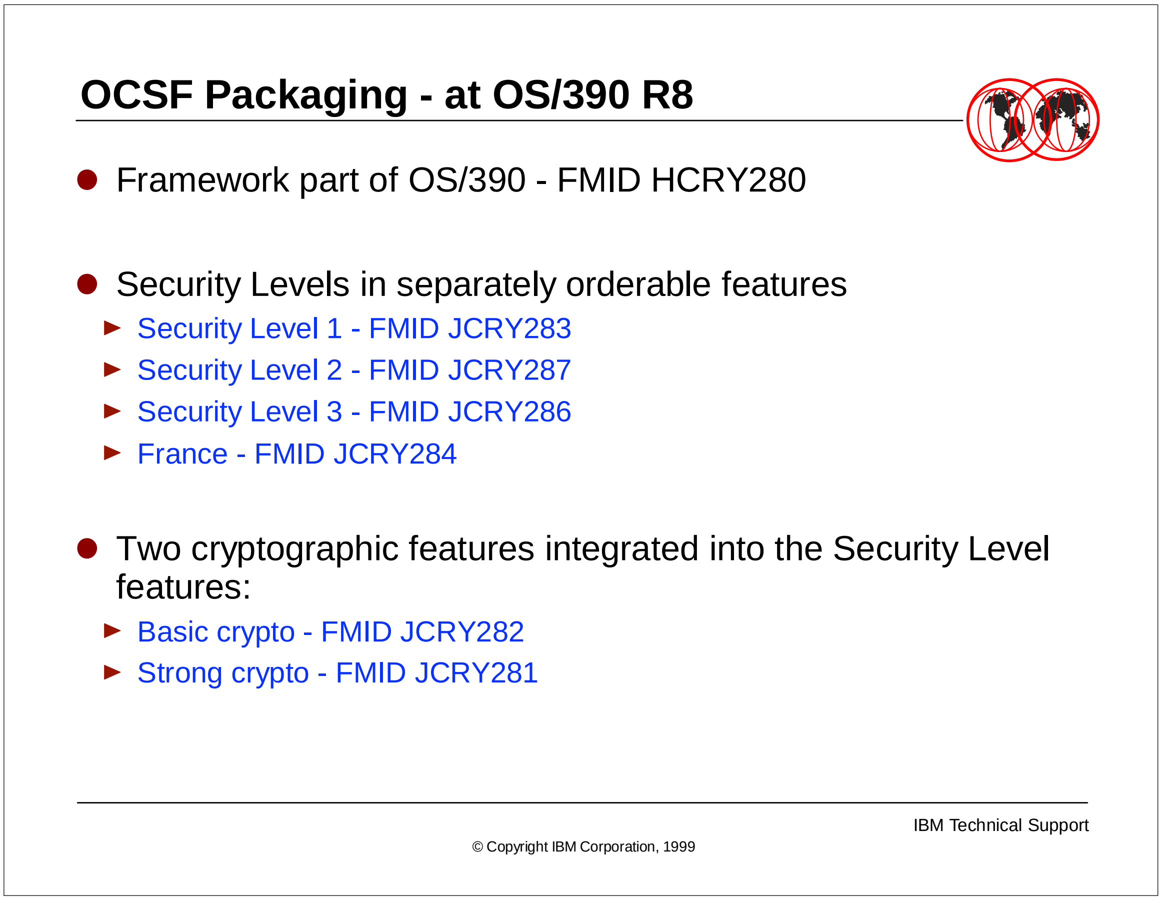
OCSF Packaging - at OS/390 R7

- Framework part of OS/390 - FMID HCRY270

- Security Levels in separately orderable features
  - Security Level 1 - FMID JCRY273
  - Security Level 2 - FMID JCRY277
  - Security Level 3 - FMID JCRY276
  - France - FMID JCRY274

- Two cryptographic features integrated into the Security Level features:
  - Basic crypto - FMID JCRY272
  - Strong crypto - FMID JCRY271

© Copyright IBM Corporation, 1999



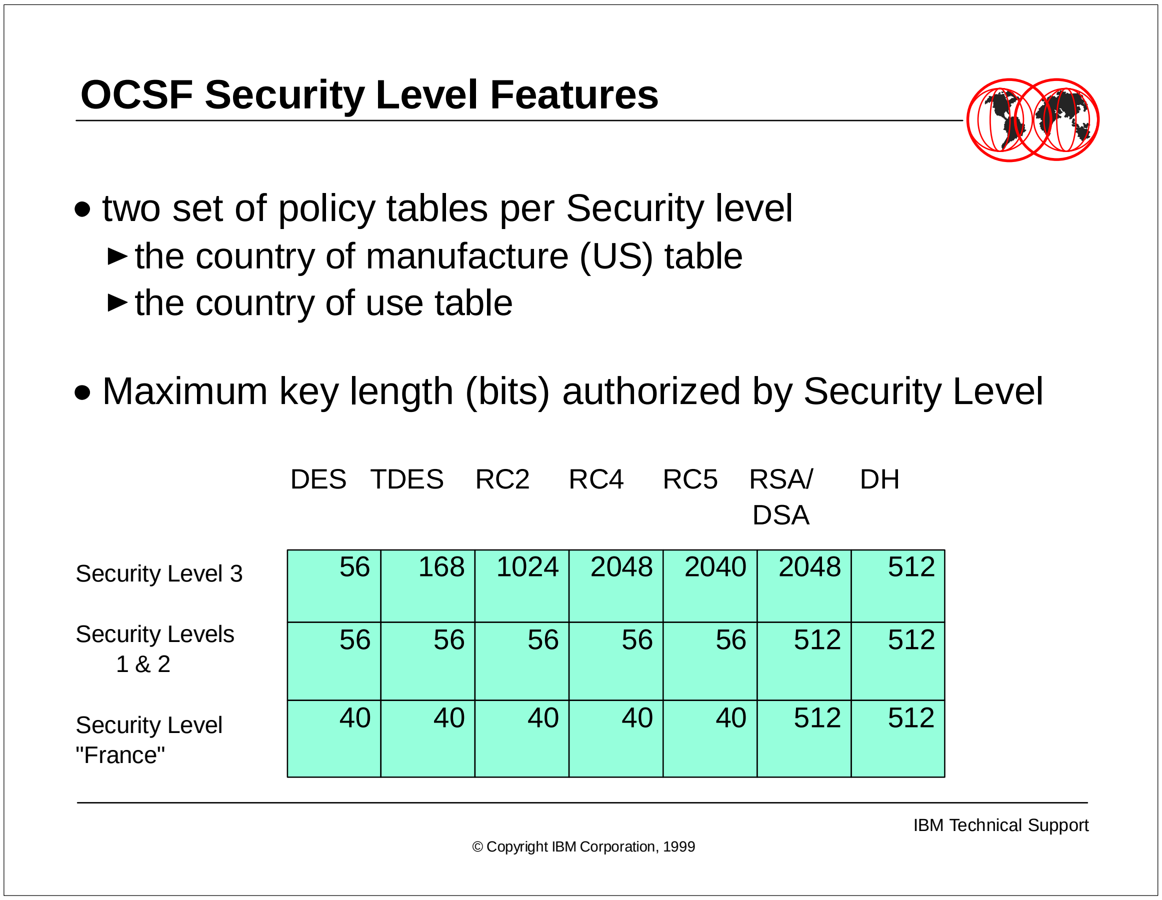
OCSF Packaging - at OS/390 R8

- Framework part of OS/390 - FMID HCRY280
- Security Levels in separately orderable features
  - Security Level 1 - FMID JCRY283
  - Security Level 2 - FMID JCRY287
  - Security Level 3 - FMID JCRY286
  - France - FMID JCRY284
- Two cryptographic features integrated into the Security Level features:
  - Basic crypto - FMID JCRY282
  - Strong crypto - FMID JCRY281

IBM Technical Support

© Copyright IBM Corporation, 1999

- The OCSF Framework is part of the base OS/390 since OS/390 R7.
- The export/control of the IBM cryptographic service providers is achieved by ordering the proper Security Level feature.
- An OCSF Security Level feature contains
  - The OCSF service providers for Trust Policy, Data Library and Certificate Library as described previously, except for the OCEP service providers.
  - The CCA Cryptographic service provider, and depending on the security level:
  - the weak software CSP alone
  - both the weak and standard software CSPs
- OCSF "policy tables" which are enforcing the encryption strength authorized by the selected Security Level.



OCSF Security Level Features

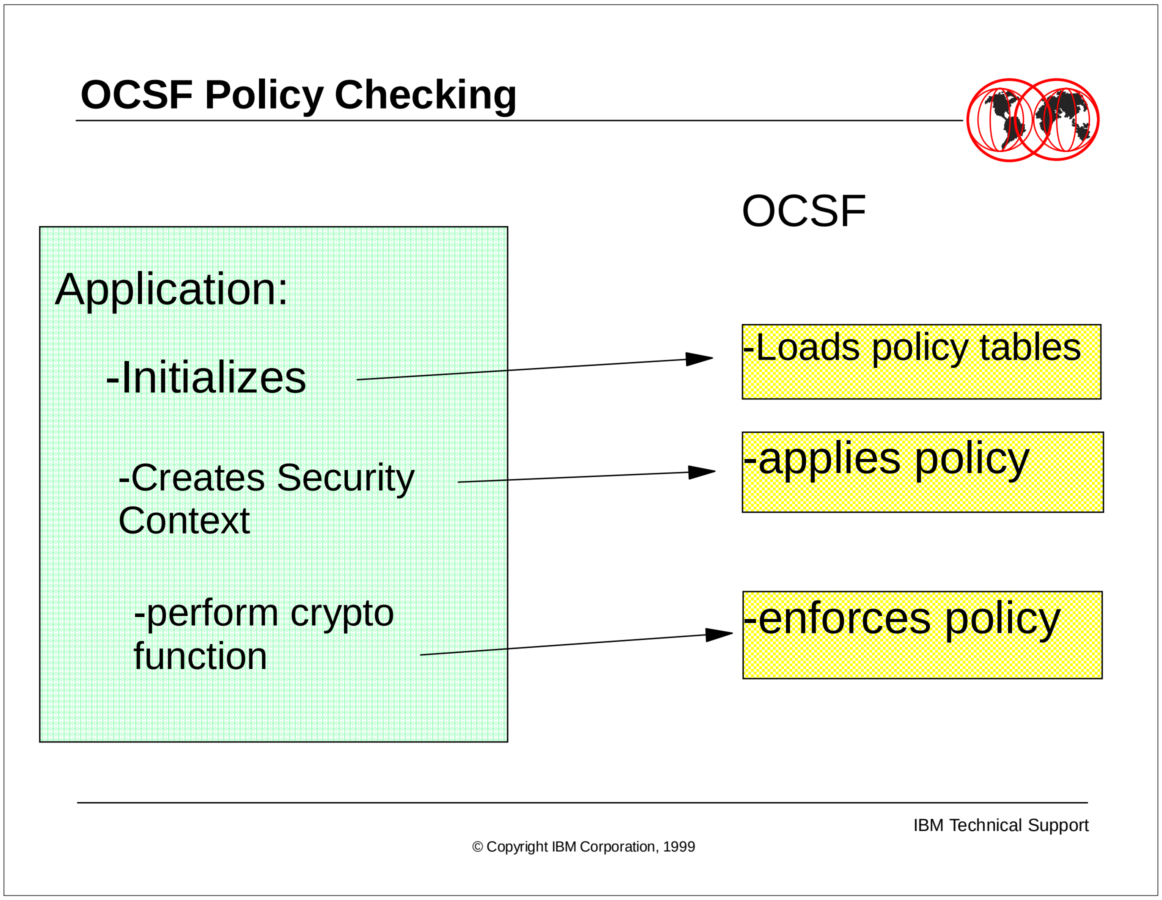
- two set of policy tables per Security level
  - ► the country of manufacture (US) table
  - ► the country of use table
- Maximum key length (bits) authorized by Security Level

	DES	TDES	RC2	RC4	RC5	RSA/ DSA	DH
Security Level 3	56	168	1024	2048	2040	2048	512
Security Levels 1 & 2	56	56	56	56	56	512	512
Security Level "France"	40	40	40	40	40	512	512

IBM Technical Support

© Copyright IBM Corporation, 1999

- ▸ For each Security Level, a set of two policy table is shipped:
  - ▸ One table for the country of manufacture, that is the US, which controls the exportation of the cryptographic algoriyhms.
  - ▸ One table for the country of use, which controls the importations restrictions on top of exportation limitations.
- ▸ The policy tables provide a list of cryptographic algorithms and their strength. A cryptographic service request for an algorithm not in the tables, or with a strength greater than specified in the tables is denied.



OCSF Policy Checking

OCSF

Application:

-Initializes

-Creates Security Context

-perform crypto function

-Loads policy tables

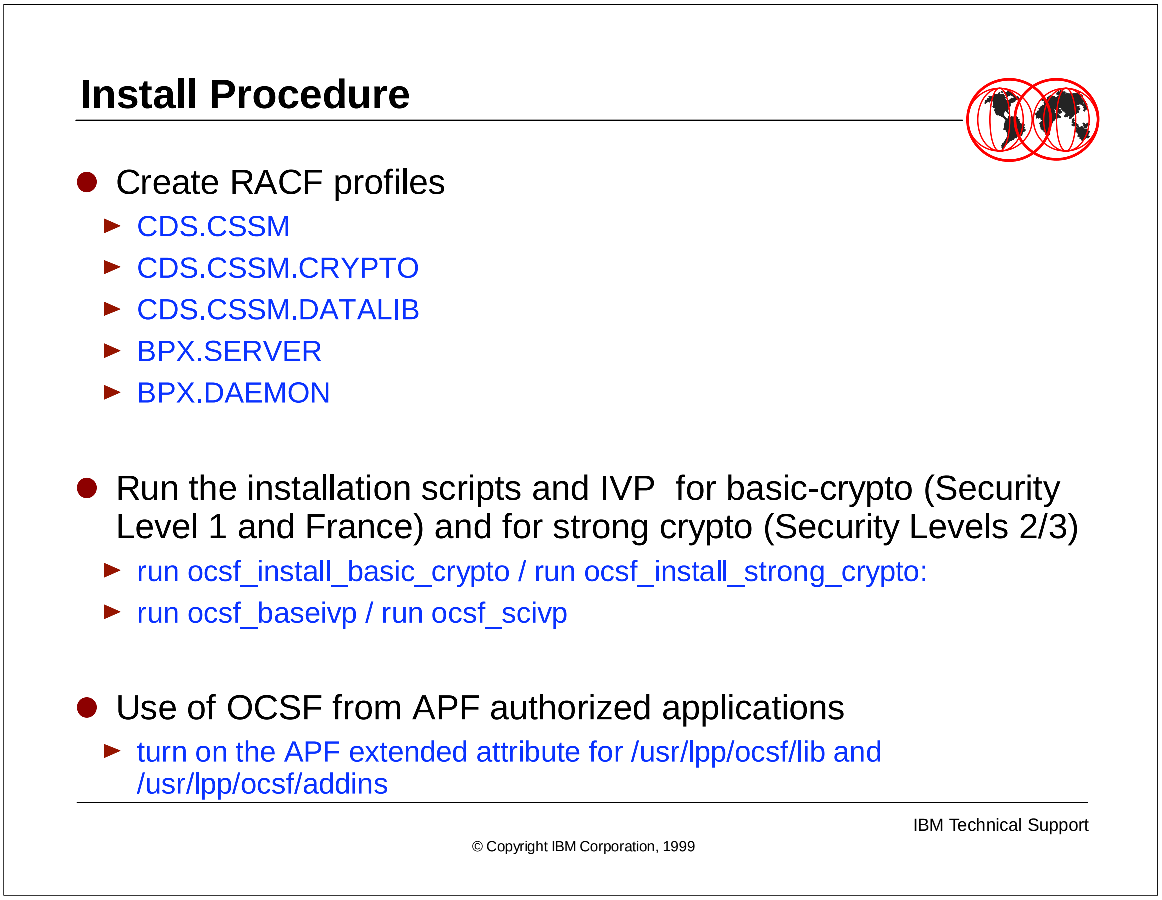
-applies policy

-enforces policy

IBM Technical Support

© Copyright IBM Corporation, 1999

- The OCSF Framework is to respond to applications requests in agreement with the policy, that is the Security Level installed.
- When the application initializes the OCSF Framework API, the OCSF Framework loads the policy tables contained in the Security Level functional modules. There is an integrity checking of the policy tables performed during this operation.l.
- As the application prepares the cryptographic operation by requesting creation of the proper "security context", that is the input/output data structure to be used between the application and the cryptographic service provider, the OCSF Framework verifies that the input parameters are matching what the installed Security Level authorizes. If the security context calls for a non authorized or unknown algorithm, the security context is flagged by the OCSF Framework to be denied if further processing has to be carried over.
- As the application actually requests the cryptographic function, the OCSF Framework checks the security context for the possible deny flag.



Install Procedure

- Create RACF profiles
  - CDS.CSSM
  - CDS.CSSM.CRYPTO
  - CDS.CSSM.DATALIB
  - BPX.SERVER
  - BPX.DAEMON
- Run the installation scripts and IVP for basic-crypto (Security Level 1 and France) and for strong crypto (Security Levels 2/3)
  - run ocsf_install_basic_crypto / run ocsf_install_strong_crypto:
  - run ocsf_baseivp / run ocsf_scivp
- Use of OCSF from APF authorized applications
  - turn on the APF extended attribute for /usr/lpp/ocsf/lib and /usr/lpp/ocsf/addins

IBM Technical Support

© Copyright IBM Corporation, 1999

- RACF profiles in the FACILITY class have to be defined to give authorizations to the OCSF daemon (that is the OCSF daemon RACF userid must be permitted for READ to these profiles):
  - CDS.CSSM - to get access to the OCSF services.
  - CDS.CSSM.CRYPTO - to get access to the cryptographic service providers services.
  - CDS.CSSM.DATALIB - to get access to the Data Library service providers services.
  - Additionally, implementing OS/390 UNIX security by defining and permitting the OCSF daemon to BPX.SERVER and BPX.DAEMON is recommended.
- Installation scripts and IVPs are provided. They are used to install the IBM OCSF service providers (except OCEP) and, depending on the Security Level installed either the weak software CSP or the weak and standard software CSPs.
- For Security Level 1 or France: run the ocsf_install_basic_crypto from /usr/lpp/ocsf/bin. This will install the service providers, the weak software CSP and the CCA CSP.
- For Security Level 2 or 3: run the ocsf_install_basic_crypto to install the service providers, the CCA CSP and the weak software CSP. In addition run ocsf_install_strong_crypto to install the standard software CSP.
- Proceed with running IVP ocsf_baseivp from /usr/lpp/ocsf/ivp, for Security Level 1 and France. baseivp checks for the correct installation of all service providers. For

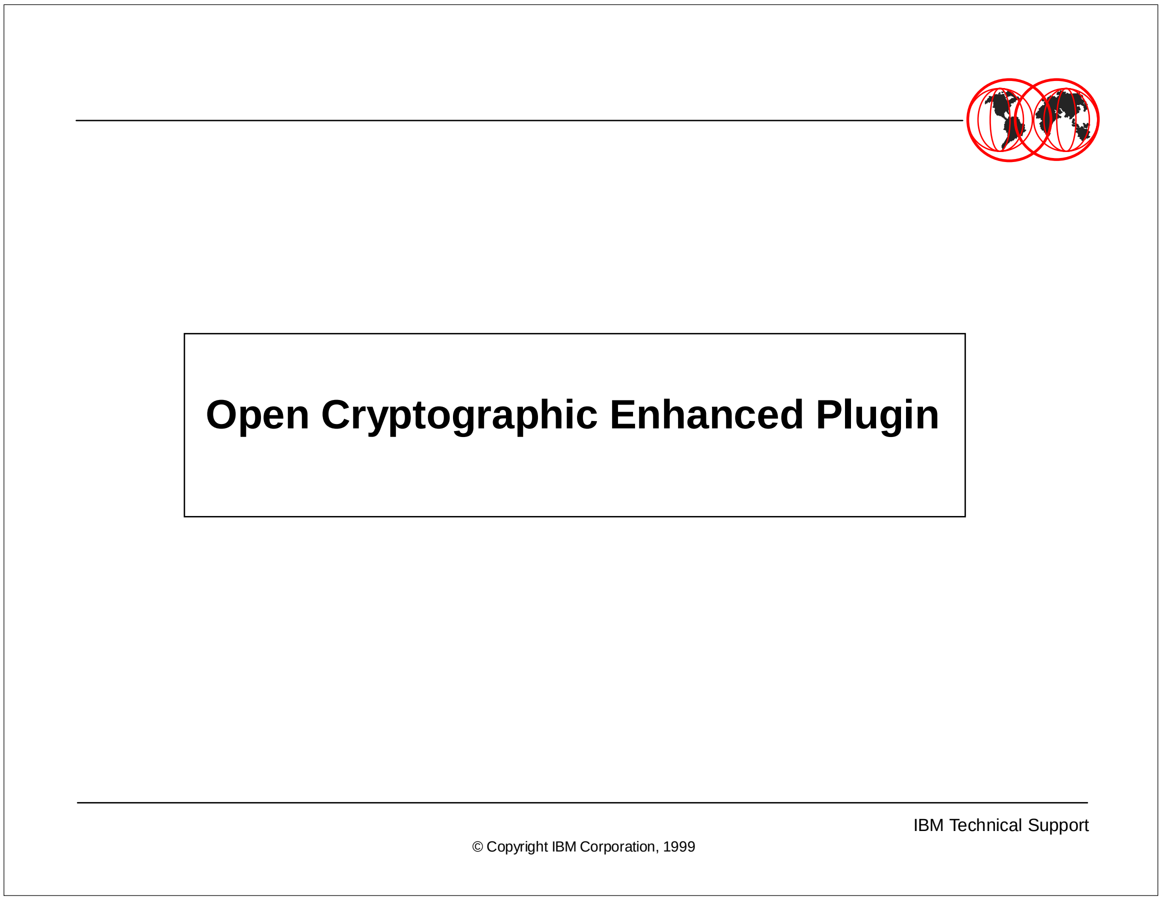
Publications

- Open Cryptographic Services Facility Developer's Guide and Reference, SC24-5875-00

- Open Cryptographic Services Facility Service Provider Guide and Reference, SC24-5876-00

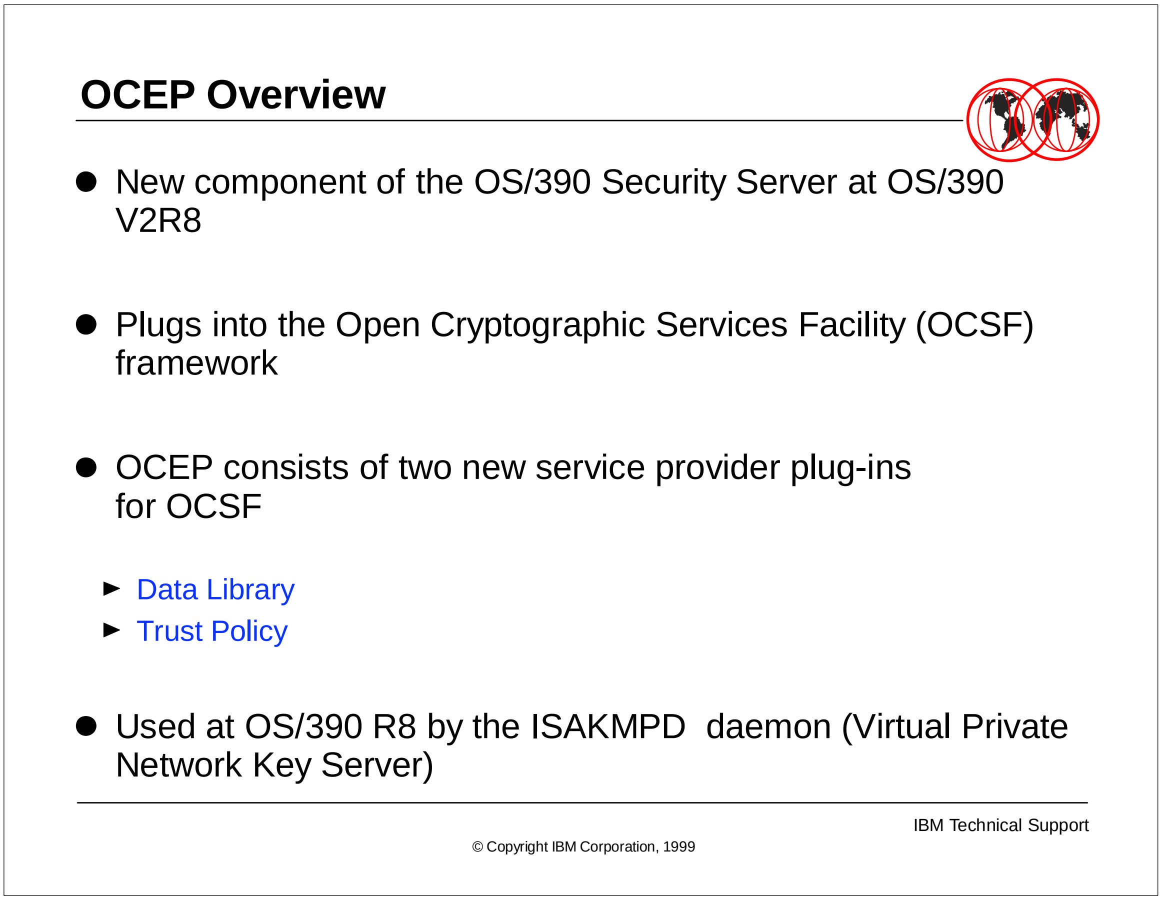
- www.opengroup.org for CDSA standards information and official data structures and APIs

© Copyright IBM Corporation, 1999



Open Cryptographic Enhanced Plugin

© Copyright IBM Corporation, 1999



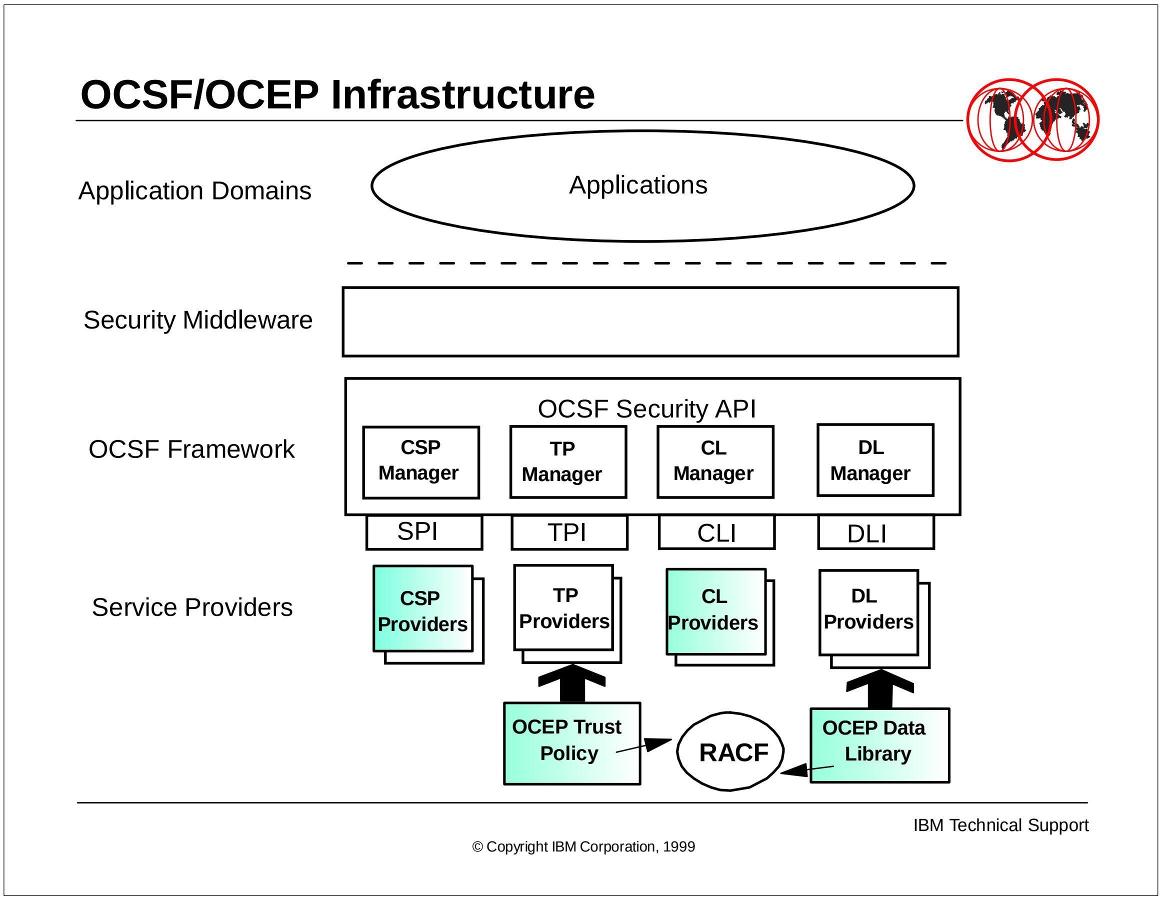
OCEP Overview

- New component of the OS/390 Security Server at OS/390 V2R8
- Plugs into the Open Cryptographic Services Facility (OCSF) framework
- OCEP consists of two new service provider plug-ins for OCSF
  - Data Library
  - Trust Policy
- Used at OS/390 R8 by the ISAKMPD daemon (Virtual Private Network Key Server)

IBM Technical Support

© Copyright IBM Corporation, 1999

- The Open Cryptographic Services Facility is the OS/390 Implementation of the Intel CDSA (Common Data Security Architecture) APIs.
- OCSF has been available since OS/390 R7, and has been shipped with IBM's own set of service provider plug-ins.
- Starting with OS/390 V2R8, RACF further contributes to the installation public key infrastructure policy. OCEP gives the OCSF users access to the facility built into RACF at this release:
  - utilization of the RACF Data Base as an OCSF Data Library
  - utilization of RACF to determine the trustworthiness of a certificate (Trust Policy).



OCSF/OCEP Infrastructure

Application Domains

Applications

Security Middleware

OCSF Framework

OCSF Security API

CSP Manager

TP Manager

CL Manager

DL Manager

SPI

TPI

CLI

DLI

Service Providers

CSP Providers

TP Providers

CL Providers

DL Providers

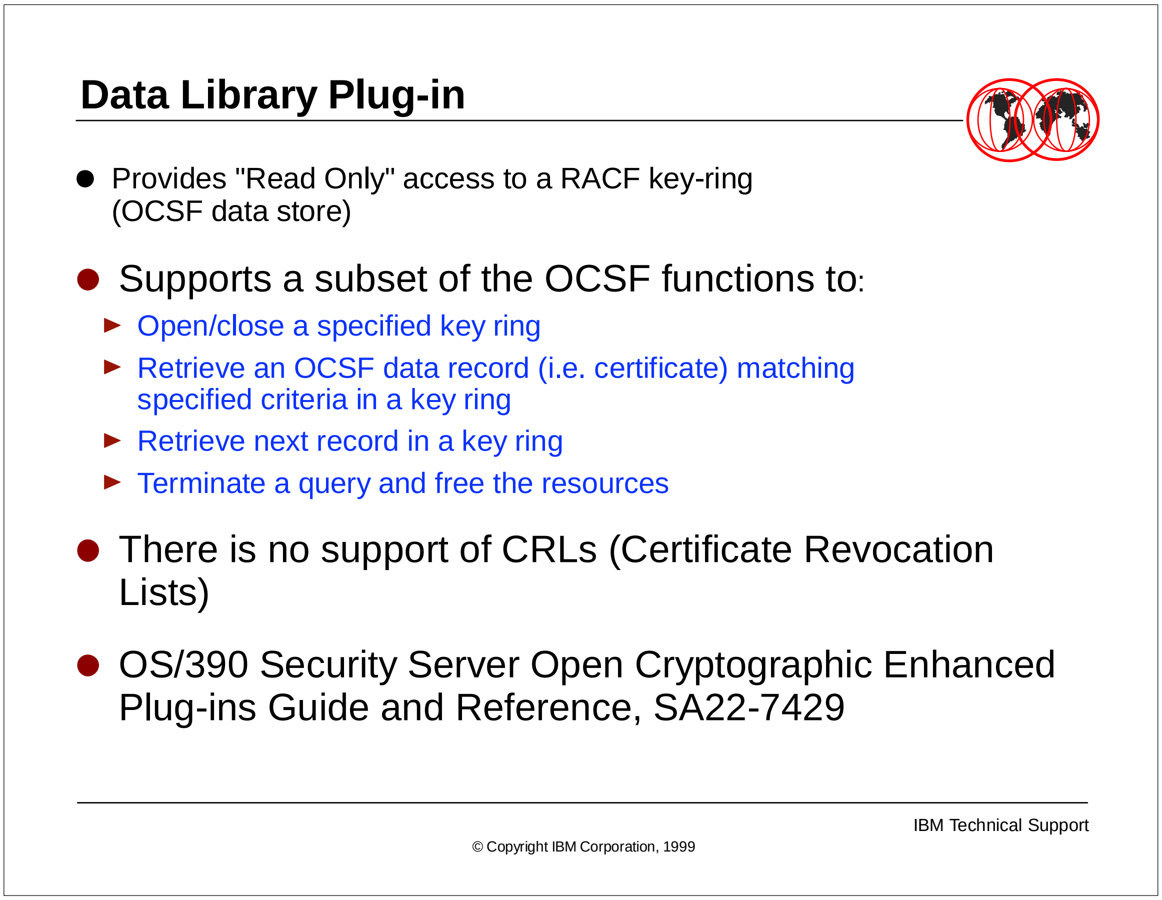
OCEP Trust Policy

RACF

OCEP Data Library

© Copyright IBM Corporation, 1999

- Here is a reminder of the infrastructure of OCSF, showing the several layers that make the APIs.
- The services which are requested by the applications, or the middleware, at the top, are performed by "pluggable" service provider modules at the bottom, the OCSF Framework acting as the logical connector between the application and the service provider.
- The services identified by CDSA are of four kinds:
    - Cryptographic services
    - Trust Policy services
    - Certificate Library services
    - Data Library services
- OCEP provides two plug-ins that come in addition to the already existing Data Library and Trust Policy modules provided by IBM, or by any other vendor providing OS/390 OCSF service provider plug-ins. The application has to explicitly request for the OCEP or non-OCEP plug-ins.

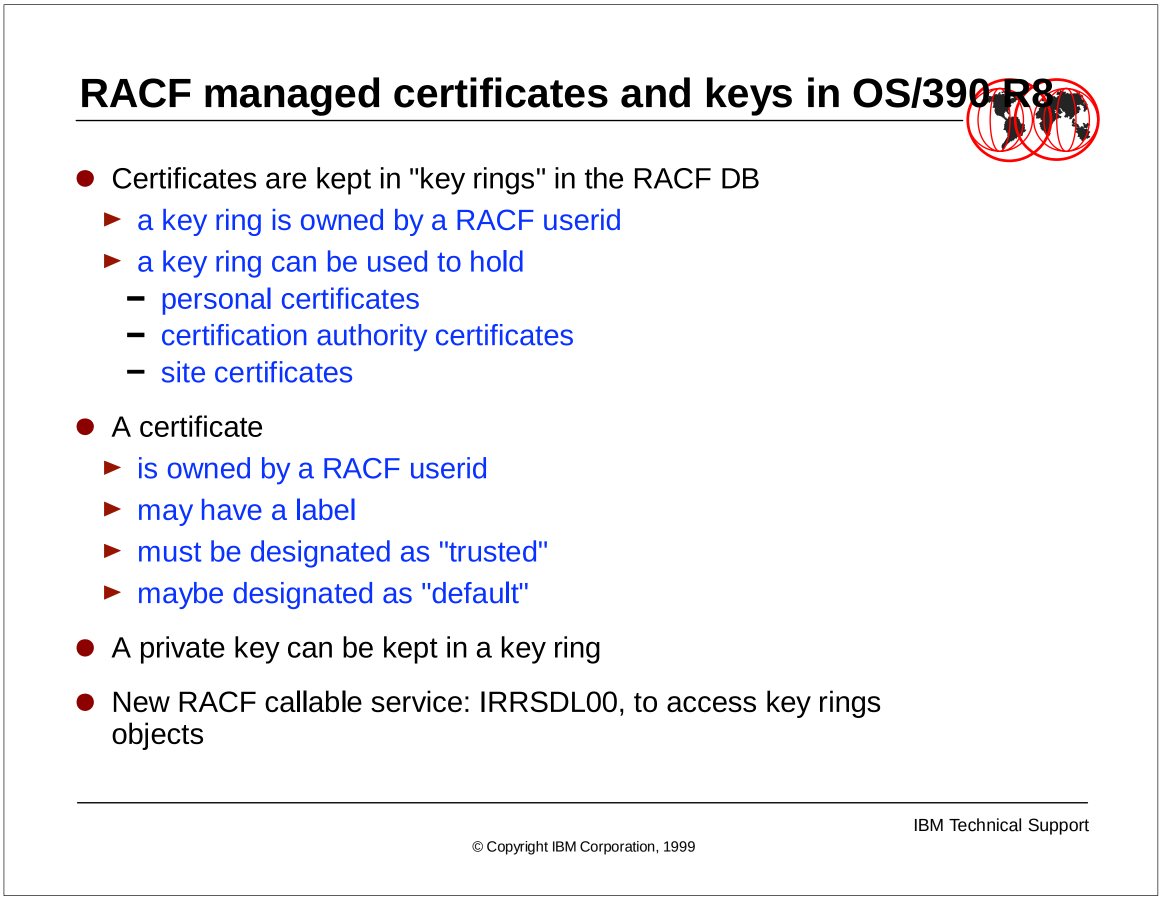


Data Library Plug-in

- Provides "Read Only" access to a RACF key-ring (OCSF data store)
- Supports a subset of the OCSF functions to:
  - Open/close a specified key ring
  - Retrieve an OCSF data record (i.e. certificate) matching specified criteria in a key ring
  - Retrieve next record in a key ring
  - Terminate a query and free the resources
- There is no support of CRLs (Certificate Revocation Lists)
- OS/390 Security Server Open Cryptographic Enhanced Plug-ins Guide and Reference, SA22-7429

© Copyright IBM Corporation, 1999

- The key rings in the RACF database appear as "data store" for OCSF through OCEP, that is persistent storage where certificates and keys can be retrieved from. There is however a global restriction in the services provided by the OCEP Data Library plug-in, in that certificates and keys can only been read from the RACF key rings.
- Another restriction is on the amount of functions available from the API. This is only a subset of the total functions that an OCSF Data Library module can provide.
- Note also that, at OS/390 R8, RACF is not able to handle certification revocation lists, which is another reason for not providing the complete set of OCSF Data Library functions.



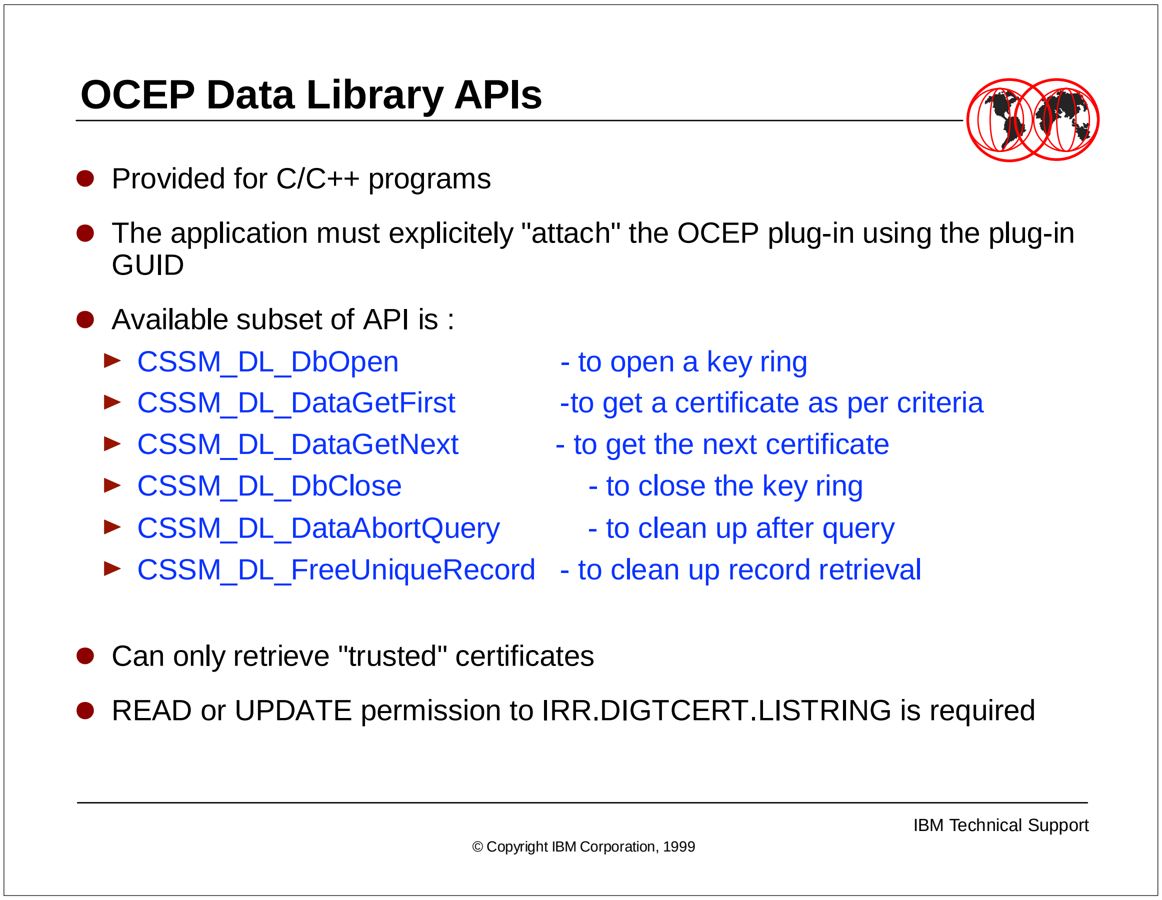
RACF managed certificates and keys in OS/390 R8

- Certificates are kept in "key rings" in the RACF DB
  - a key ring is owned by a RACF userid
  - a key ring can be used to hold
    - personal certificates
    - certification authority certificates
    - site certificates
- A certificate
  - is owned by a RACF userid
  - may have a label
  - must be designated as "trusted"
  - maybe designated as "default"
- A private key can be kept in a key ring
- New RACF callable service: IRRSDL00, to access key rings objects

IBM Technical Support

© Copyright IBM Corporation, 1999

- Certificates are actually records in the RACF Database. Corresponding private keys, if present, can be stored also as records in the RACF Database or as records in the ICSF PKDS (Public Key Data Set).
- Users can create key rings in the RACF Database, which are actually pointers to the certificates and private keys records.
- Certificates, keys and key rings are owned by RACF userids.



OCEP Data Library APIs

- Provided for C/C++ programs
- The application must explicitely "attach" the OCEP plug-in using the plug-in GUID
- Available subset of API is :

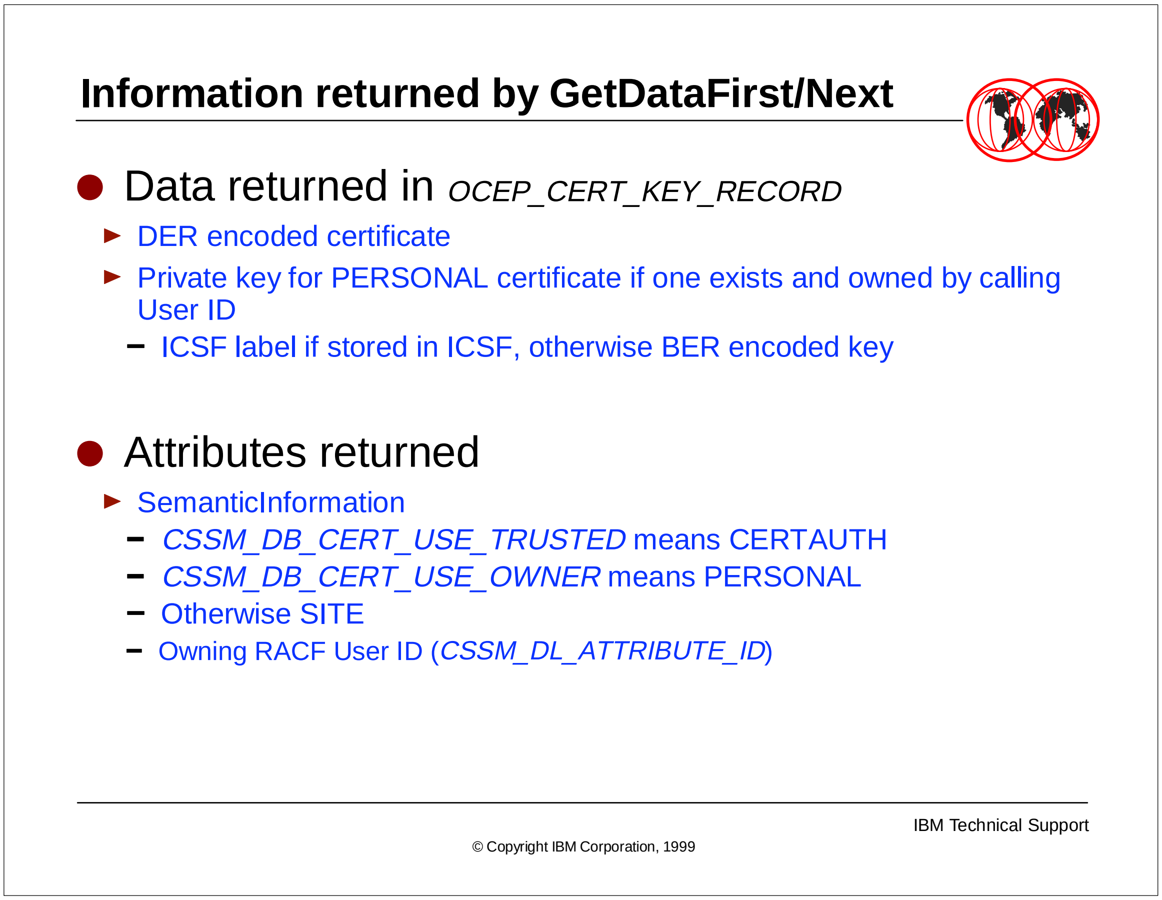
► CSSM_DL_DbOpen	- to open a key ring
► CSSM_DL_DataGetFirst	-to get a certificate as per criteria
► CSSM_DL_DataGetNext	- to get the next certificate
► CSSM_DL_DbClose	- to close the key ring
► CSSM_DL_DataAbortQuery	- to clean up after query
► CSSM_DL_FreeUniqueRecord	- to clean up record retrieval

- Can only retrieve "trusted" certificates
- READ or UPDATE permission to IRR.DIGTCERT.LISTRING is required

IBM Technical Support

© Copyright IBM Corporation, 1999

- OCSF provides APIs for C/C++
- When an application wants to use a service provider, it requests to the CSSM framework to be “attached” to the specific service provider..
- OCSF most of the time uses the CDSA terminology, which needs to be mapped to the OS/390 implementation of CDSA concepts. As an example what CDSA refers to as a "data store" translates into a RACF "key ring".
- Here are the names of the functions made available for C/C++ to call the OCEP Data library services.
- the access to the functions are controlled by the permission the accessing userid has to the IRR.DIGTCERT.LISTRING facility class profile (at least READ permission is required).



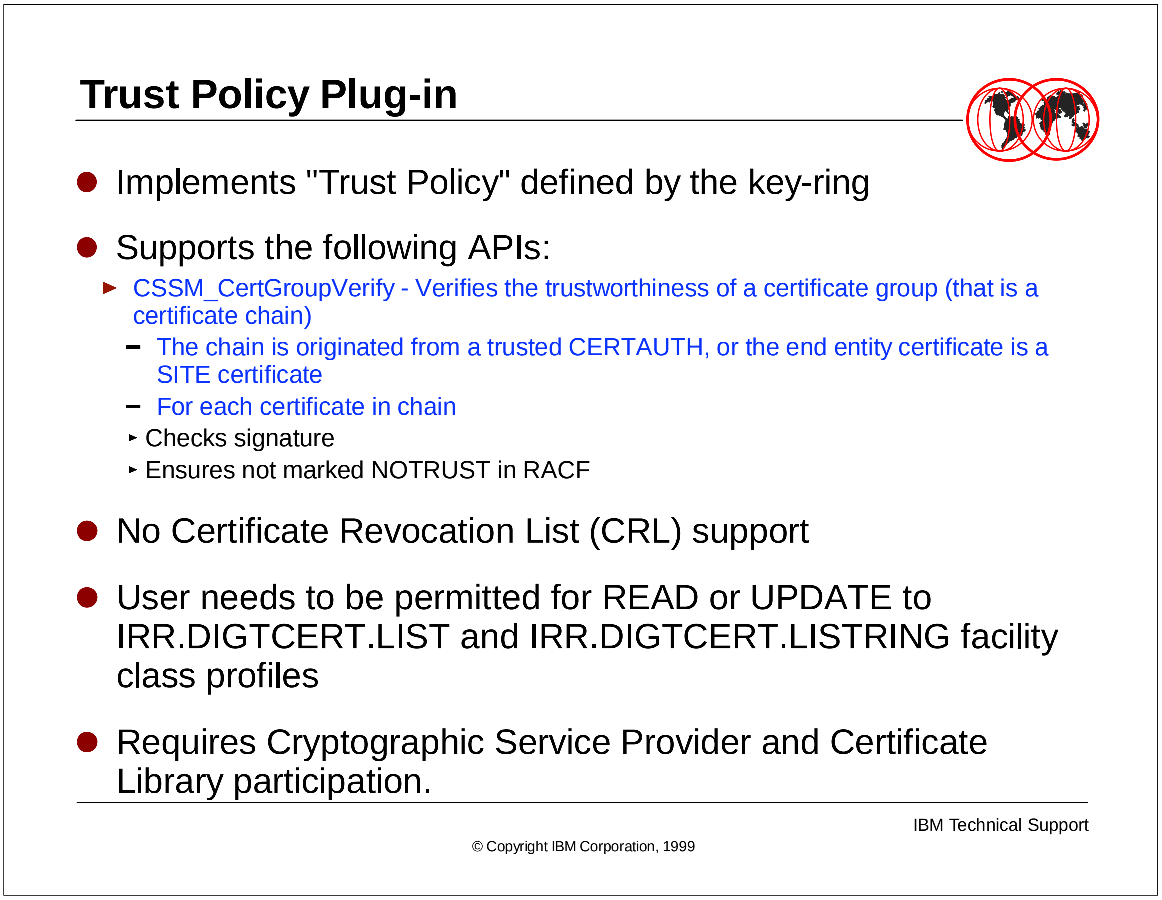
Information returned by GetDataFirst/Next

- Data returned in *OCEP_CERT_KEY_RECORD*
  - DER encoded certificate
  - Private key for PERSONAL certificate if one exists and owned by calling User ID
    - ICSF label if stored in ICSF, otherwise BER encoded key
- Attributes returned
  - SemanticInformation
    - *CSSM_DB_CERT_USE_TRUSTED* means CERTAUTH
    - *CSSM_DB_CERT_USE_OWNER* means PERSONAL
    - Otherwise SITE
    - Owning RACF User ID (*CSSM_DL_ATTRIBUTE_ID*)

IBM Technical Support

© Copyright IBM Corporation, 1999

- Here are the information returned to the application by the OCEP Data Library service provider. As mentioned before, these are only a subset of the potential capabilities provided by OCSF. Nevertheless it provides a good view of what the CDSA Data Library concept is intended for.



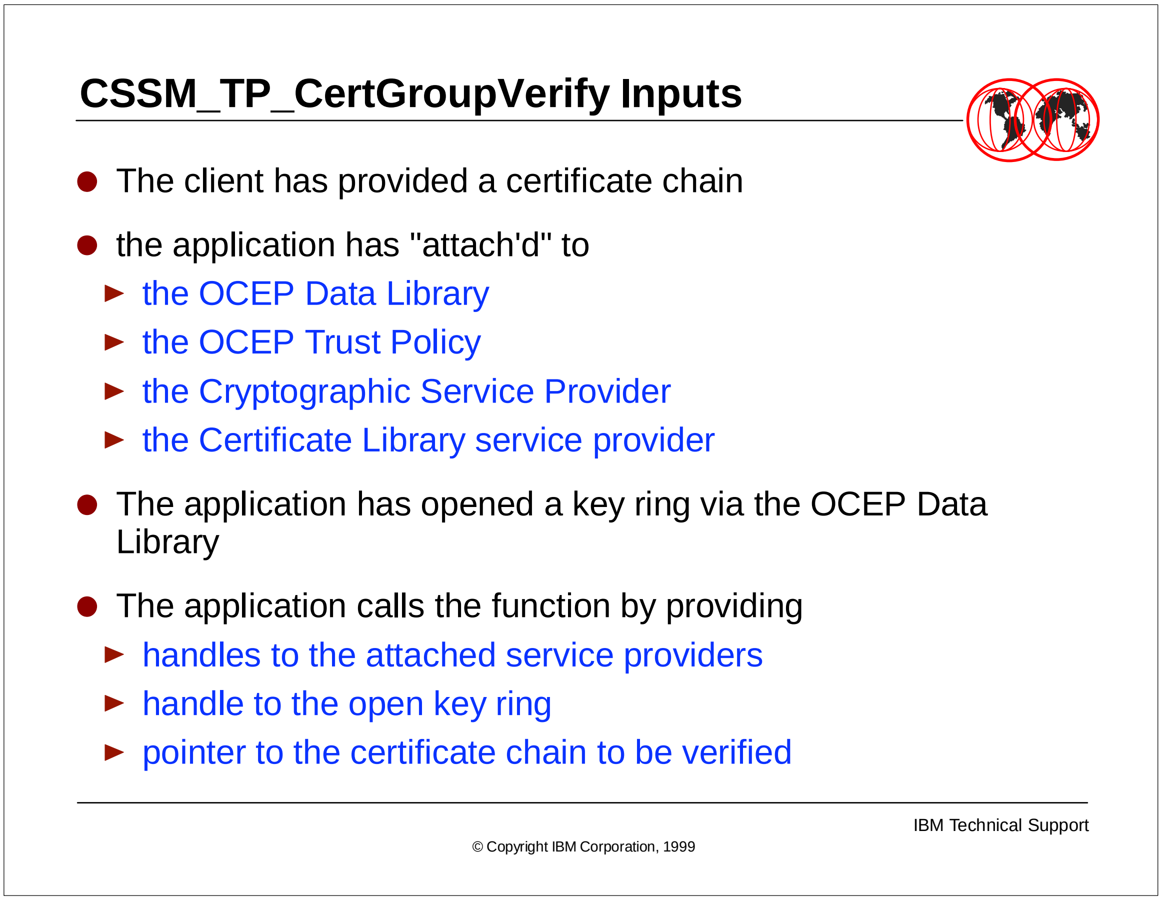
Trust Policy Plug-in

- Implements "Trust Policy" defined by the key-ring
- Supports the following APIs:
  - CSSM_CertGroupVerify - Verifies the trustworthiness of a certificate group (that is a certificate chain)
    - The chain is originated from a trusted CERTAUTH, or the end entity certificate is a SITE certificate
    - For each certificate in chain
    - Checks signature
    - Ensures not marked NOTRUST in RACF
- No Certificate Revocation List (CRL) support
- User needs to be permitted for READ or UPDATE to IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING facility class profiles
- Requires Cryptographic Service Provider and Certificate Library participation.

IBM Technical Support

© Copyright IBM Corporation, 1999

- The OCEP Trust Policy service provider is also to use the information stored in the RACF Database, with the objective of establishing the validity of a certificate as per the information stored in a key ring.
- As for the OCEP Data Library plug-in, only a subset of the OCSF Trust Policy APIs is supported. This is due to the current constraints in the RACF Database implementation, where, for example, CRLs cannot be handled.
- The trust policy needs participation of the cryptographic service provider to perform the service it proposes. This is done under the cover.



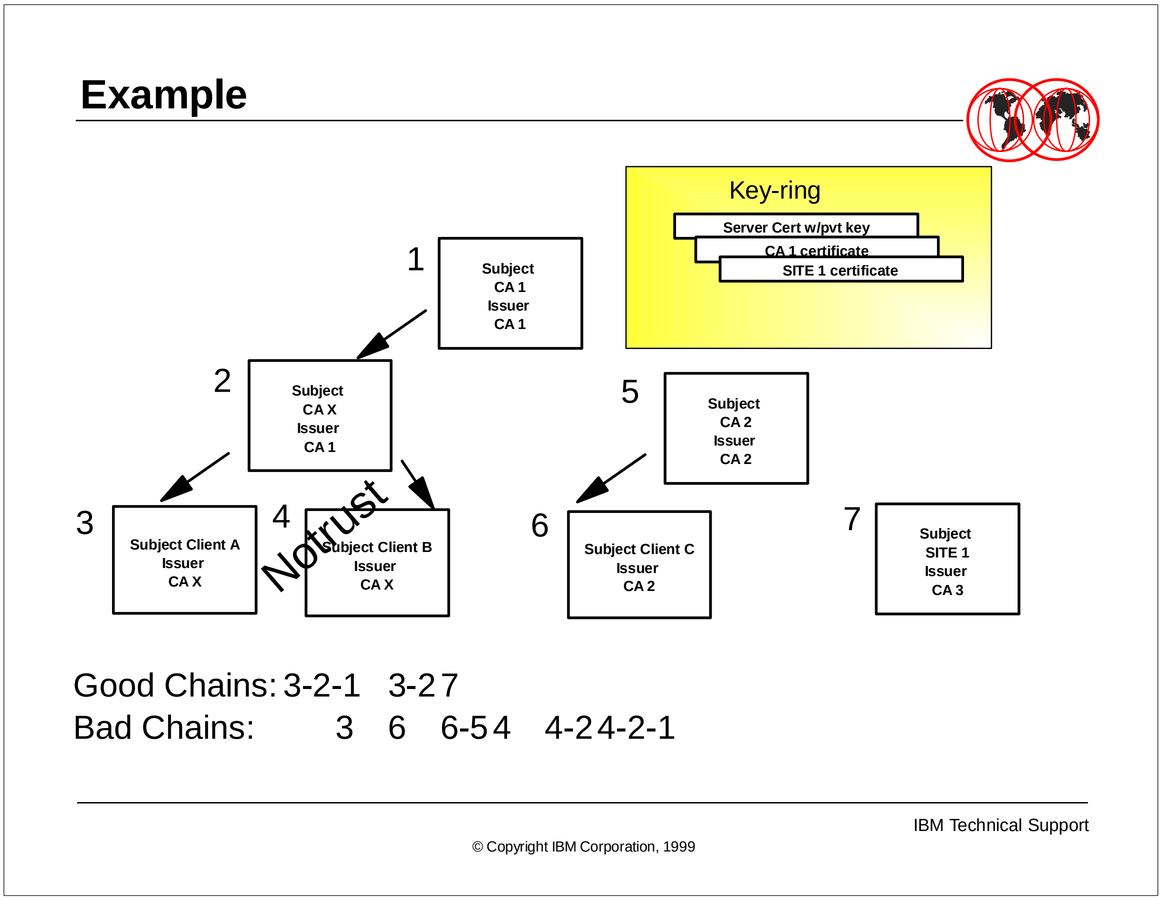
CSSM_TP_CertGroupVerify Inputs

- The client has provided a certificate chain
- the application has "attach'd" to
  - the OCEP Data Library
  - the OCEP Trust Policy
  - the Cryptographic Service Provider
  - the Certificate Library service provider
- The application has opened a key ring via the OCEP Data Library
- The application calls the function by providing
  - handles to the attached service providers
  - handle to the open key ring
  - pointer to the certificate chain to be verified

IBM Technical Support

© Copyright IBM Corporation, 1999

- This provides a good overview of the typical mode of operation of OCSF, where a service is to be given using multiple service providers, whereas the application is actually interfacing with only one provider for this specific service. Service providers are calling each other using also the CSSM Framework.
- Note that the two OCEP service providers are contributing, in that access to the RACF key ring containing the CERTAUTH and/or SITE certificates to be used for the chain verification can only be performed through the OCEP Data Library plug-in.



Example

Key-ring

Server Cert w/pvt key

CA 1 certificate

SITE 1 certificate

1 Subject CA 1 Issuer CA 1

2 Subject CA X Issuer CA 1

3 Subject Client A Issuer CA X

4 Subject Client B Issuer CA X

Notrust

5 Subject CA 2 Issuer CA 2

6 Subject Client C Issuer CA 2

7 Subject SITE 1 Issuer CA 3

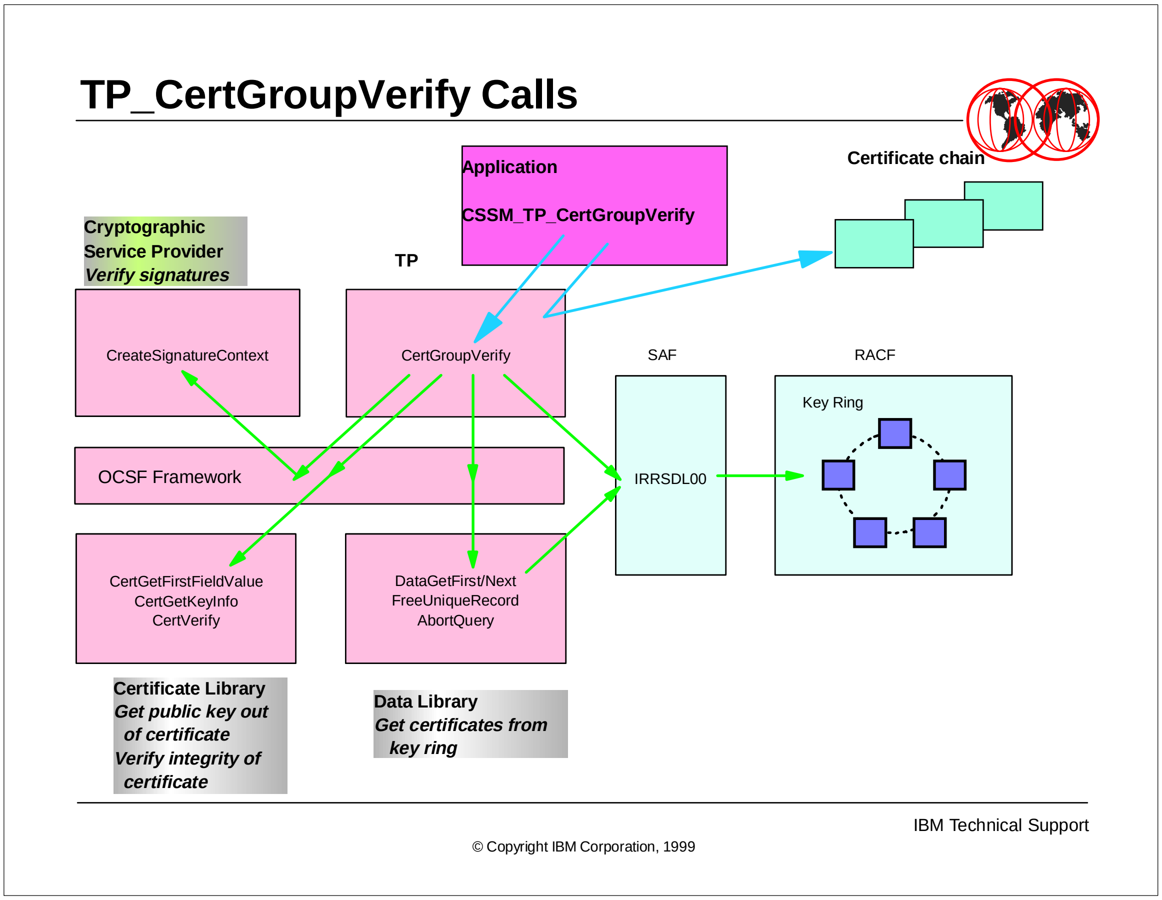
Good Chains: 3-2-1   3-2 7

Bad Chains: 3   6   6-5 4   4-2 4-2-1

IBM Technical Support

© Copyright IBM Corporation, 1999

- In this example we are considering 4 chains of certificates as delivered by a client that the application wants to verify:
  - chain 3-2-1
  - chain 4-2-1
  - chain 6-5
  - chain 7, which is a single certificate.
  - Note that all the certificates in these chains have been signed by certification authorities.
- We are describing how these chains are verified, as a result of calling the CSSM_CerGroupVerify function, against the key ring shown at the top of the foil.



TP_CertGroupVerify Calls

Application

CSSM_TP_CertGroupVerify

Certificate chain

Cryptographic
Service Provider
Verify signatures

TP

CreateSignatureContext

CertGroupVerify

SAF

RACF

OCSF Framework

IRRSDL00

Key Ring

CertGetFirstFieldValue
CertGetKeyInfo
CertVerify

DataGetFirst/Next
FreeUniqueRecord
AbortQuery

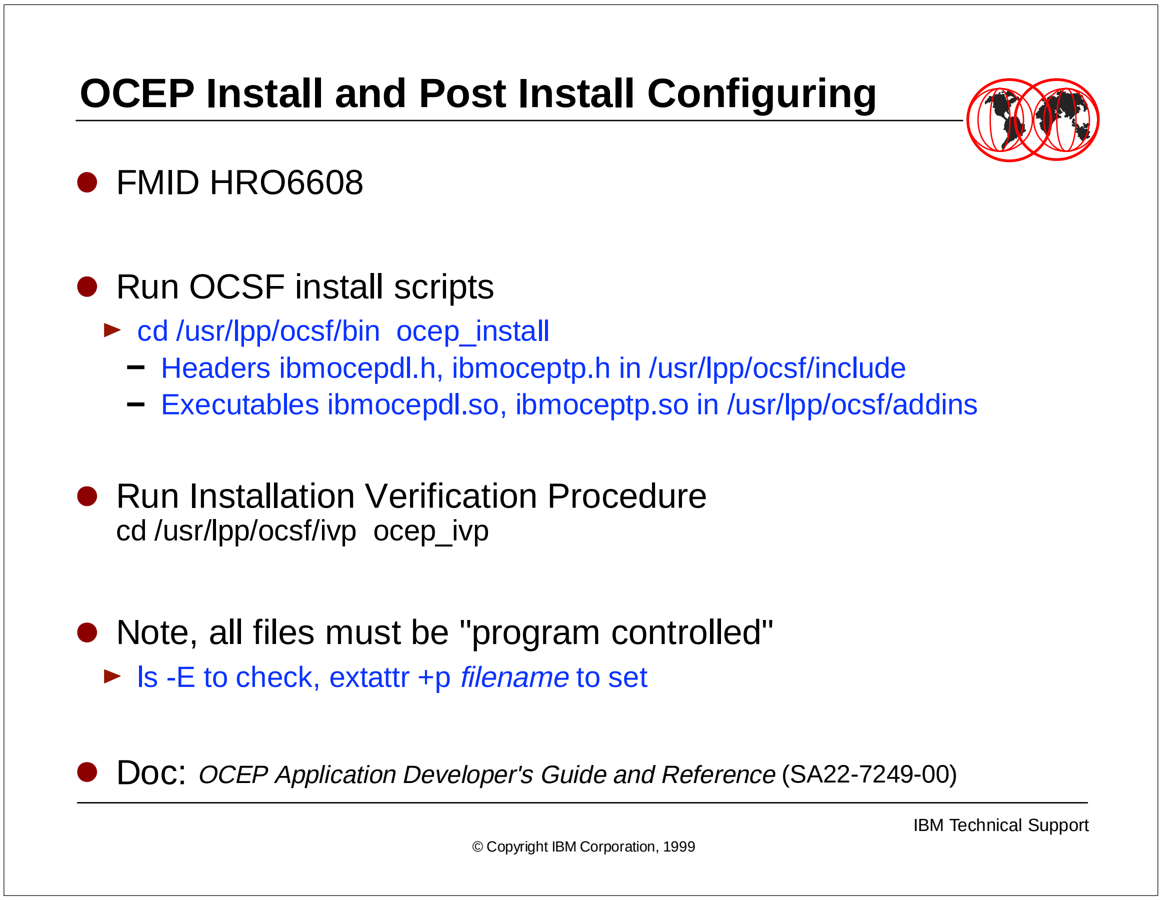
Certificate Library
Get public key out of certificate
Verify integrity of certificate

Data Library
Get certificates from key ring

IBM Technical Support

© Copyright IBM Corporation, 1999

- This is an overall view of the interactions occurring between the OCSF service providers when using the OCEP Trust Policy
- The OCSF Framework is controlling these interactions
- The OCEP service providers use the IRRSDL00 services to address requests to RACF.



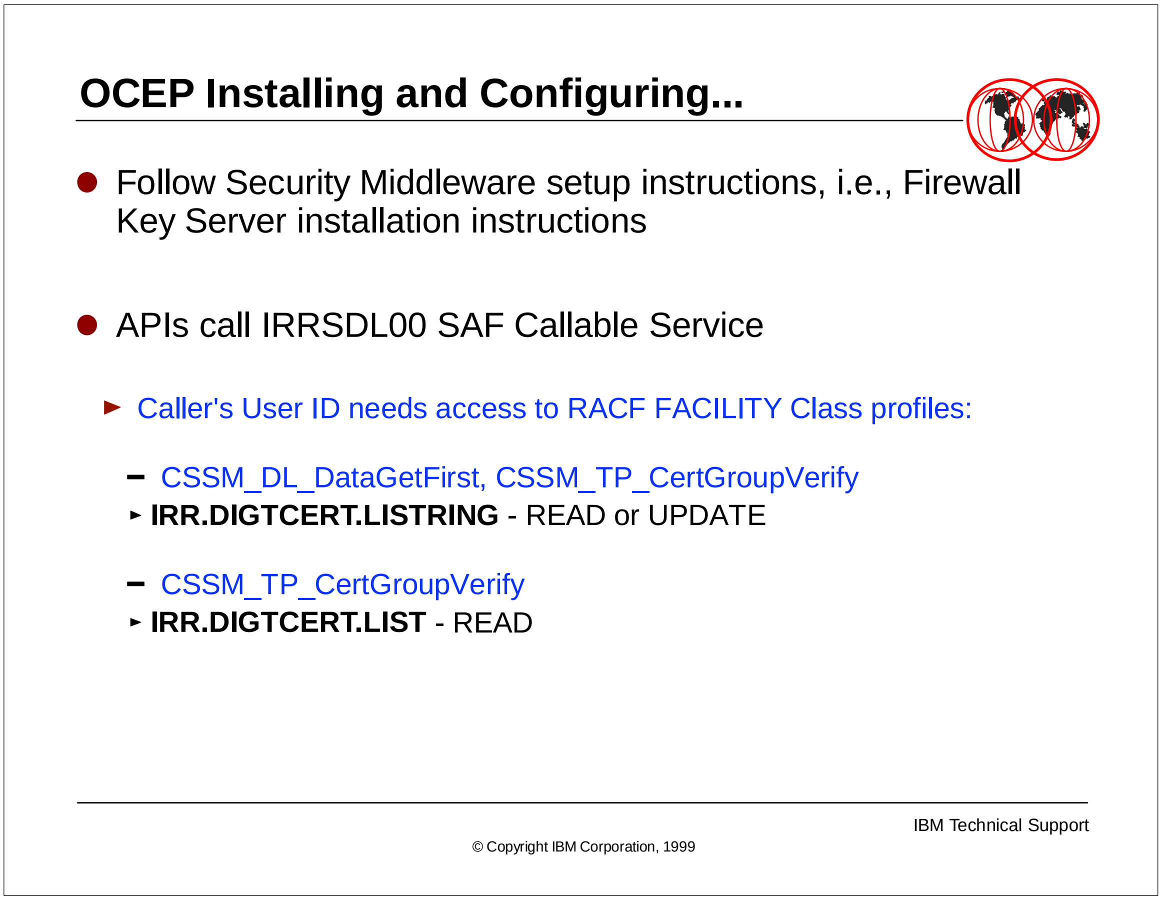
OCEP Install and Post Install Configuring

- FMID HRO6608
- Run OCSF install scripts
  - cd /usr/lpp/ocsf/bin ocep_install
    - Headers ibmocepdl.h, ibmoceptp.h in /usr/lpp/ocsf/include
    - Executables ibmocepdl.so, ibmoceptp.so in /usr/lpp/ocsf/addins
- Run Installation Verification Procedure
  cd /usr/lpp/ocsf/ivp ocep_ivp
- Note, all files must be "program controlled"
  - ls -E to check, extattr +p *filename* to set
- Doc: *OCEP Application Developer's Guide and Reference* (SA22-7249-00)

IBM Technical Support

© Copyright IBM Corporation, 1999

- OCSF must be up and running before installing the OCEP Plug-ins
- OCEP itself is installed by the ocep_install script, which also registers OCEP with the OCSF Framework.
- OCEP comes with an installation verification procedure script.
- The installation and verification scripts are run from the OS/390 UNIX shell.
- There is a requirement that all OCEP executable files be program controlled.



OCEP Installing and Configuring...

- Follow Security Middleware setup instructions, i.e., Firewall Key Server installation instructions
- APIs call IRRSDL00 SAF Callable Service
  - Caller's User ID needs access to RACF FACILITY Class profiles:
    - CSSM_DL_DataGetFirst, CSSM_TP_CertGroupVerify
      - **IRR.DIGTCERT.LISTRING** - READ or UPDATE
    - CSSM_TP_CertGroupVerify
      - **IRR.DIGTCERT.LIST** - READ

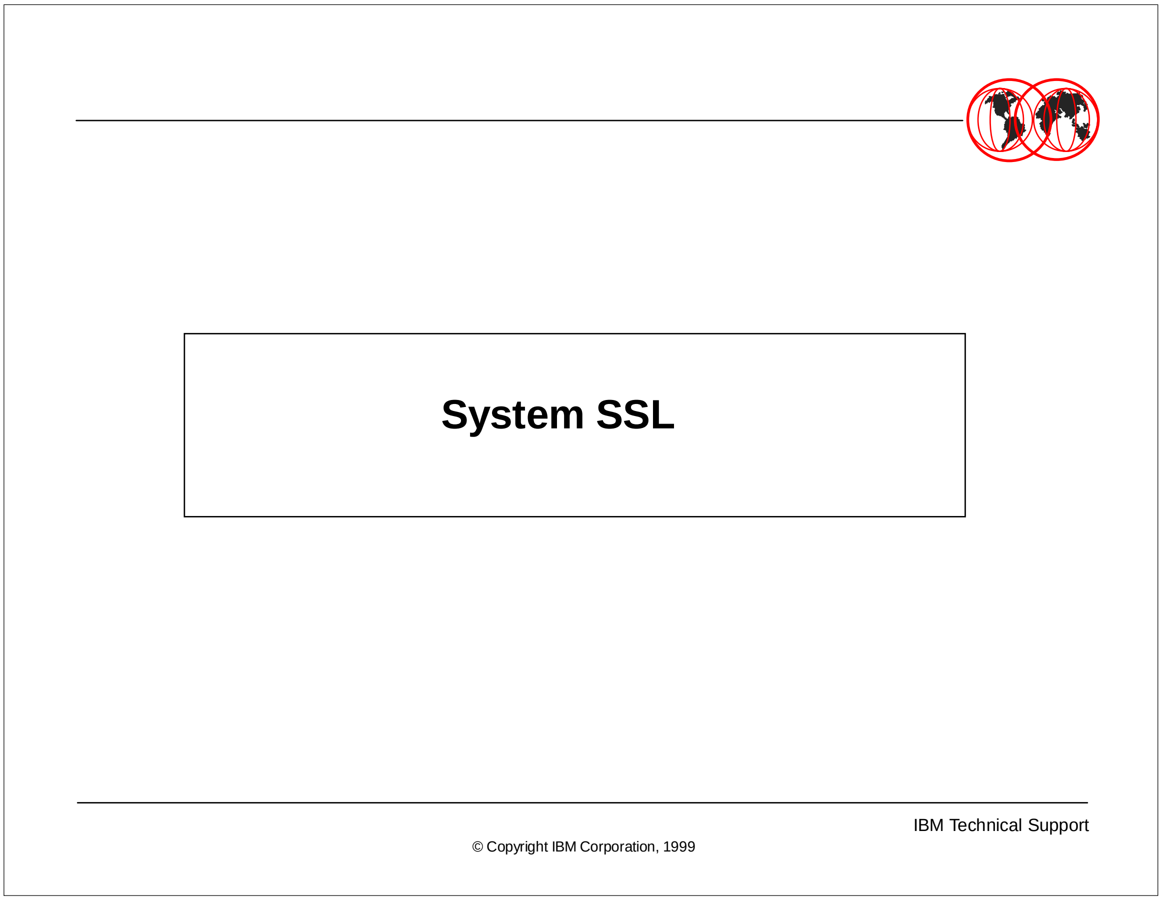
© Copyright IBM Corporation, 1999

- This shows what additional configuration is needed, with the example of the ISAKMPD daemon. to use OCEP. The daemon userid will have to be permitted to the IRR.DIGTCERT.LISTRING and IRR.DIGTCERT.LIST profiles.

References

- SA22-7249 OCEP Guide and Reference
- SC24-5875 OCSF Developer's Guide and Reference
- SC24-5876 OCSF Service Provider Module Guide and Reference
- GC28-1921 RACF Callable Services

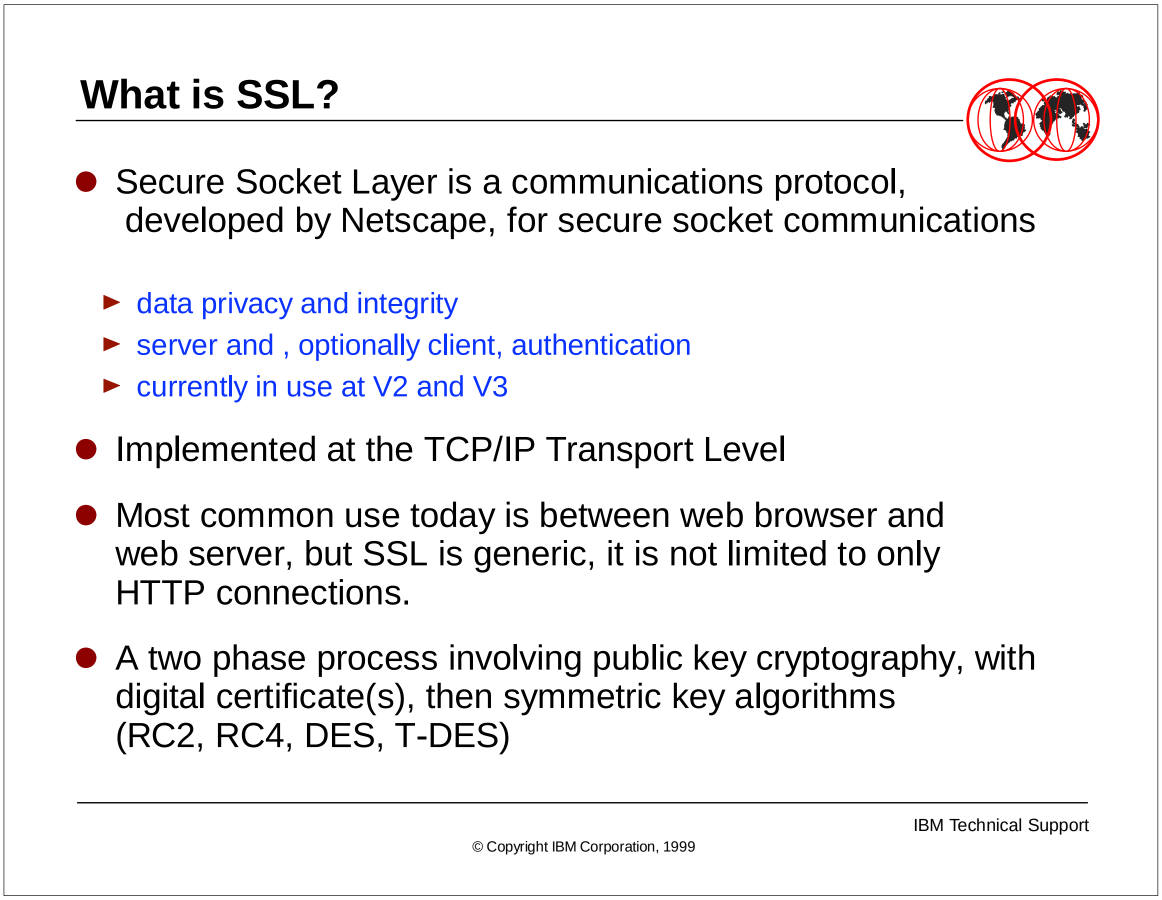
© Copyright IBM Corporation, 1999



System SSL

IBM Technical Support

© Copyright IBM Corporation, 1999



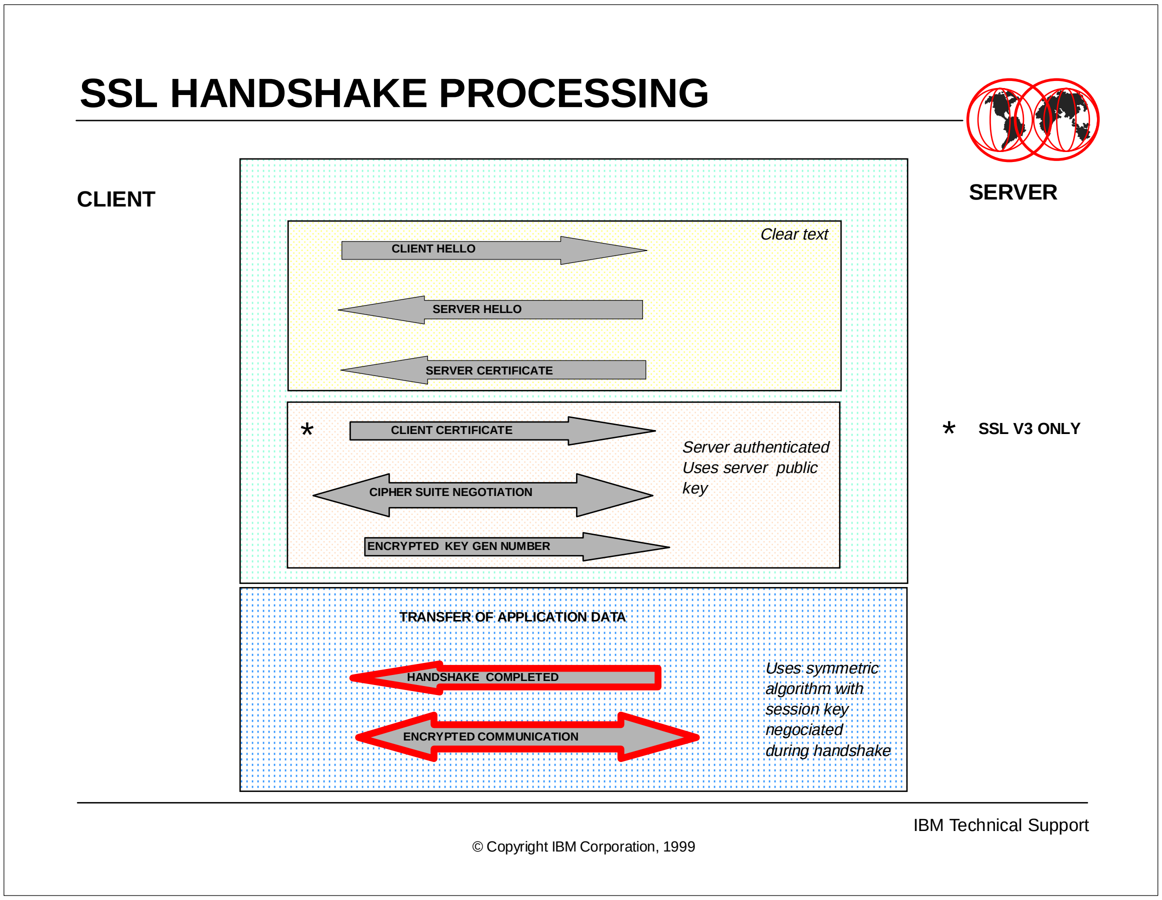
What is SSL?

- Secure Socket Layer is a communications protocol, developed by Netscape, for secure socket communications
  - data privacy and integrity
  - server and , optionally client, authentication
  - currently in use at V2 and V3
- Implemented at the TCP/IP Transport Level
- Most common use today is between web browser and web server, but SSL is generic, it is not limited to only HTTP connections.
- A two phase process involving public key cryptography, with digital certificate(s), then symmetric key algorithms (RC2, RC4, DES, T-DES)

IBM Technical Support

© Copyright IBM Corporation, 1999

- Secure Socket Layer (SSL) is a well known protocol, widely in use to secure communications between browsers and web servers.
- However the use of SLL is not restricted to the http protocol only, any programs that communicate over TCP/IP sockets could use SSL to secure their conversation.
- SSL requires an extensive use of cryptographic services, and to optimize the induced computational workload, it uses both asymmetric, or public key, algorithms (very costly in terms of computation) and the more efficient symmetric algorithms, such as DES.
- The use of public key cryptography implies using digital certificates as well.



SSL HANDSHAKE PROCESSING

CLIENT

SERVER

Clear text

CLIENT HELLO

SERVER HELLO

SERVER CERTIFICATE

* CLIENT CERTIFICATE

CIPHER SUITE NEGOTIATION

ENCRYPTED KEY GEN NUMBER

Server authenticated
Uses server public key

* SSL V3 ONLY

TRANSFER OF APPLICATION DATA

HANDSHAKE COMPLETED

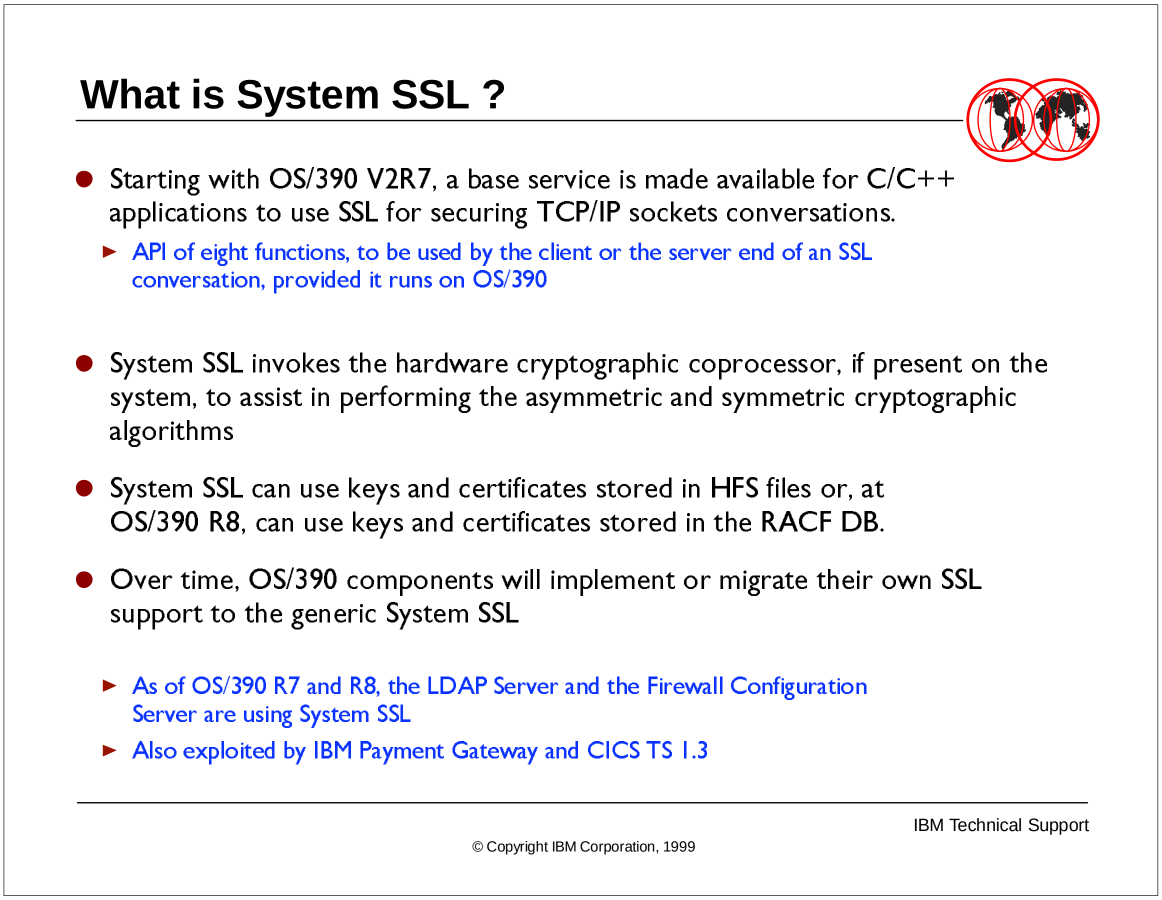
ENCRYPTED COMMUNICATION

Uses symmetric algorithm with session key negociated during handshake

IBM Technical Support

© Copyright IBM Corporation, 1999

- This foils shows, at a very high level, the interactions between a client and a server during an SSL conversation.
- The conversation is initiated by a “handshake” phase, which is used to negotiate the symmetric algorithm to use and to securely build the same shared secret key at the client and at the server.
- Note that SSL V3 is giving the opportunity to get a certificate from the client too as well during the handshake.
- Once the handshake completes, the communication is carried on using symmetric encryption and the key used during the handshake.



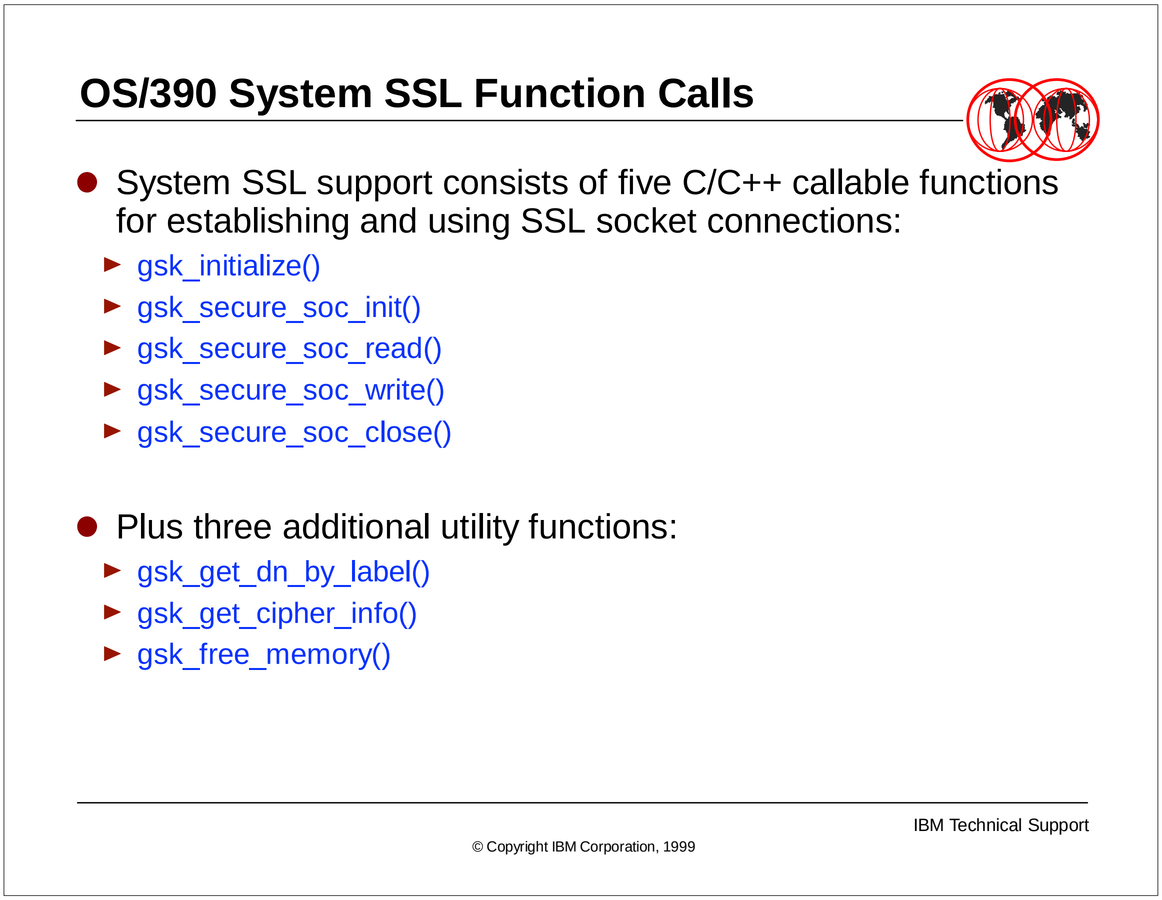
What is System SSL ?

- Starting with OS/390 V2R7, a base service is made available for C/C++ applications to use SSL for securing TCP/IP sockets conversations.
  - API of eight functions, to be used by the client or the server end of an SSL conversation, provided it runs on OS/390
- System SSL invokes the hardware cryptographic coprocessor, if present on the system, to assist in performing the asymmetric and symmetric cryptographic algorithms
- System SSL can use keys and certificates stored in HFS files or, at OS/390 R8, can use keys and certificates stored in the RACF DB.
- Over time, OS/390 components will implement or migrate their own SSL support to the generic System SSL
  - As of OS/390 R7 and R8, the LDAP Server and the Firewall Configuration Server are using System SSL
  - Also exploited by IBM Payment Gateway and CICS TS 1.3

IBM Technical Support

© Copyright IBM Corporation, 1999

- Some components of OS/390 have been using SSL for some time now. They came with their own implementation of SSL support.
- In OS/390 R7, a built in generic SSL service is available in the OS/390 base, which can be called by C/C++ applications through eight new C/C++ functions.
- New OS/390 components with SSL requirements are using the base System SSL service, other components using their own SSL service are expected to migrate to using System SSL.
- As of OS/390 R7, the Firewall configuration server and the LDAP server are the only two OS/390 Component using System SSL.
- System SSL uses the hardware cryptographic coprocessor if installed on the system.
- The use of SSL implies the use of digital certificates and of private keys. There are facilities available in OS/390 for the management of keys and certificates. They will be discussed later in this presentation.



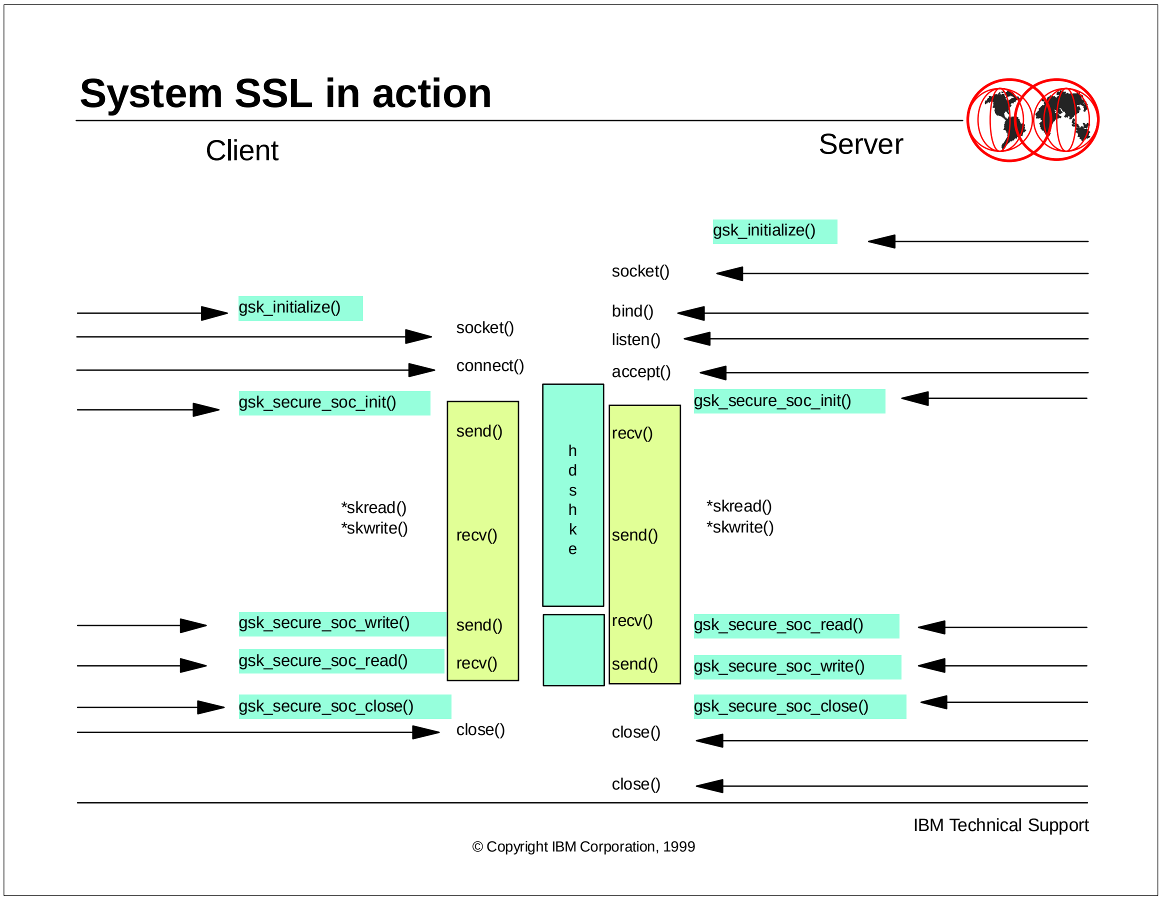
OS/390 System SSL Function Calls

- System SSL support consists of five C/C++ callable functions for establishing and using SSL socket connections:
  - gsk_initialize()
  - gsk_secure_soc_init()
  - gsk_secure_soc_read()
  - gsk_secure_soc_write()
  - gsk_secure_soc_close()
- Plus three additional utility functions:
  - gsk_get_dn_by_label()
  - gsk_get_cipher_info()
  - gsk_free_memory()

IBM Technical Support

© Copyright IBM Corporation, 1999

- Here are the System SSL API functions available for any C/C++ application starting with OS/390 R7. These functions calls System SSL to establish a socket communication and to wrap the conversation in the SSL protocol.
- Other useful utility functions are provided, that allow to dynamically manage the set of SSL related information needed by the application.



System SSL in action

Client

Server

gsk_initialize()

socket()

gsk_initialize()

bind()

socket()

listen()

connect()

accept()

gsk_secure_soc_init()

gsk_secure_soc_init()

send()

h
d
s
h
k
e

recv()

*skread()
*skwrite()

recv()

send()

*skread()
*skwrite()

gsk_secure_soc_write()

send()

recv()

gsk_secure_soc_read()

gsk_secure_soc_read()

recv()

send()

gsk_secure_soc_write()

gsk_secure_soc_close()

gsk_secure_soc_close()

close()

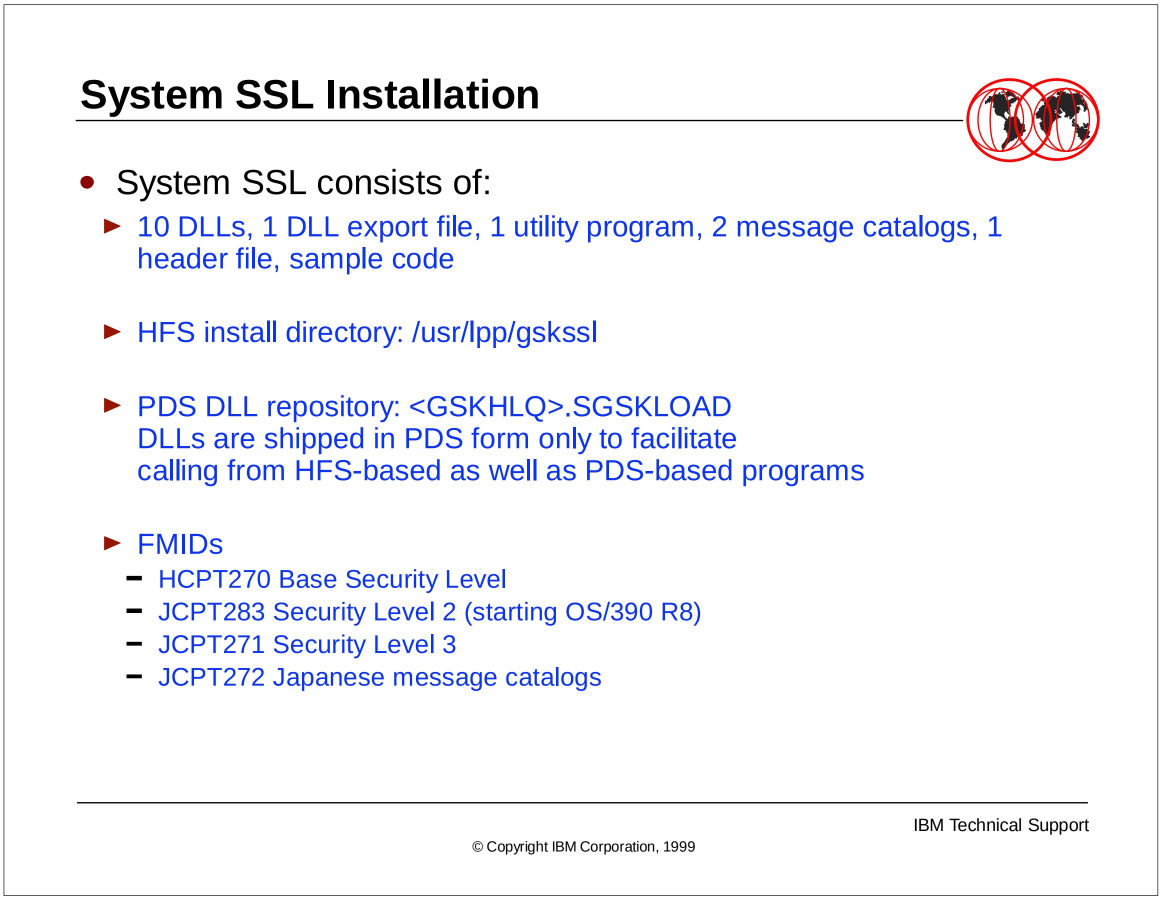
close()

close()

IBM Technical Support

© Copyright IBM Corporation, 1999

- Here is the socket programming model used with System SSL
- There is still a need to establish the communication between a client and a server via socket creation, binding and connection. As there is a need to perform send() and recv() functions to exchange data.
- The application uses the System SSL API, which in turn invoke the "SSL layer", which is taking care of all the cryptographic activities on top of the regular socket functions.

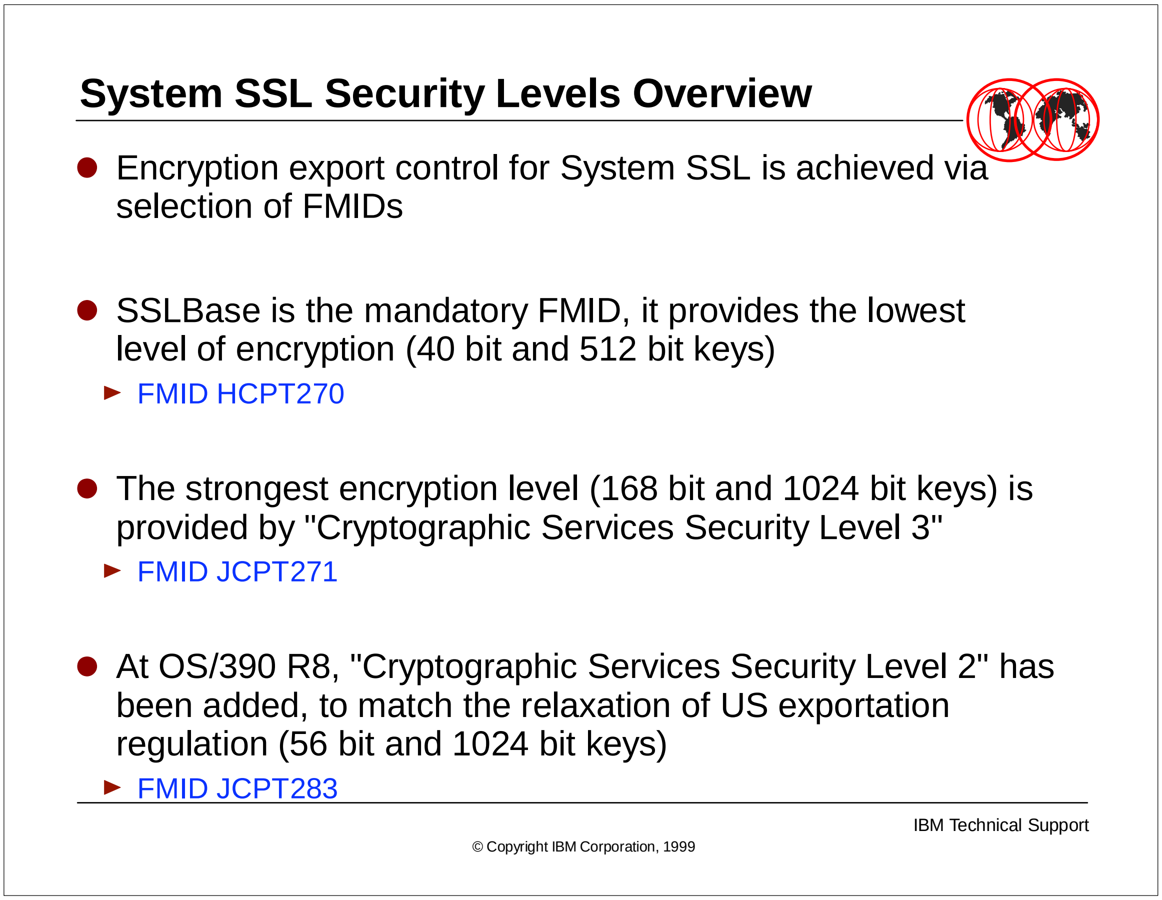


System SSL Installation

- System SSL consists of:
  - 10 DLLs, 1 DLL export file, 1 utility program, 2 message catalogs, 1 header file, sample code
  - HFS install directory: /usr/lpp/gskssl
  - PDS DLL repository: <GSKHLQ>.SGSKLOAD
    DLLs are shipped in PDS form only to facilitate
    calling from HFS-based as well as PDS-based programs
  - FMIDs
    - HCPT270 Base Security Level
    - JCPT283 Security Level 2 (starting OS/390 R8)
    - JCPT271 Security Level 3
    - JCPT272 Japanese message catalogs

© Copyright IBM Corporation, 1999

- The DLLs used by System SSL are installed in OS/390 PDS, so that both HFS executables and OS/390 load modules can invoke these DLLs.
- Additional facilities, such as header file, samples, message catalogs, and the gskkyman utility are delivered in HFS files.
- FMIDs HCPT270, JCPT283 and JCPT271 are explained in the next foil. FMID JCPT272 is the japanese message catalogs for gskkyman.

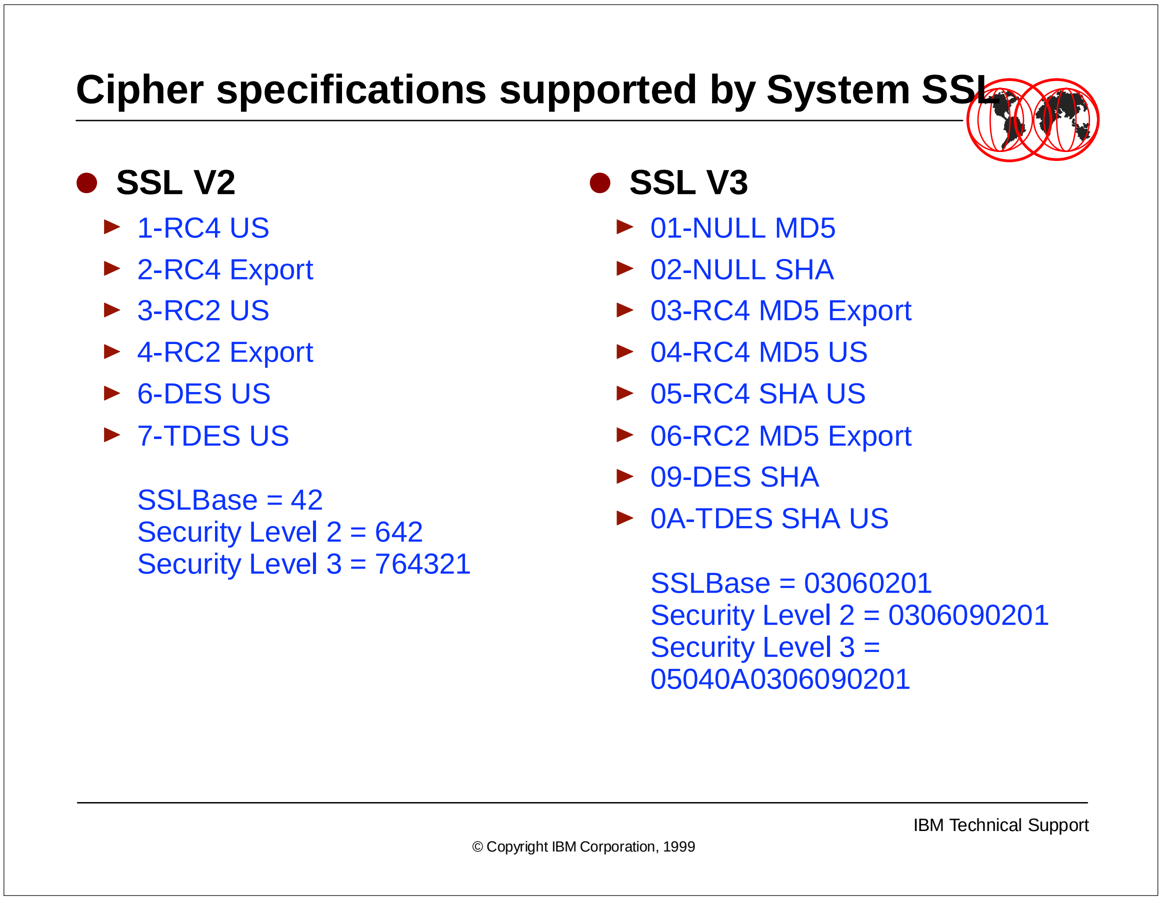


System SSL Security Levels Overview

- Encryption export control for System SSL is achieved via selection of FMIDs
- SSLBase is the mandatory FMID, it provides the lowest level of encryption (40 bit and 512 bit keys)
  - FMID HCPT270
- The strongest encryption level (168 bit and 1024 bit keys) is provided by "Cryptographic Services Security Level 3"
  - FMID JCPT271
- At OS/390 R8, "Cryptographic Services Security Level 2" has been added, to match the relaxation of US exportation regulation (56 bit and 1024 bit keys)
  - FMID JCPT283

© Copyright IBM Corporation, 1999

- The control of exportation and importation of the cryptographic algorithms built in System SSL is enforced by using different FMIDs, depending on the desired supported key length.
- OS/390 R7 offered two possible levels of encryption:
- the SSL “base level” encryption, which is the lowest encryption level, with symmetric keys such as 40 bit long.
- The SSL “domestic” level encryption, with 128 long symmetric keys or triple DES.
- OS/390 R8 introduces an intermediate “level 2", to end up now with three levels:
  - The SSL “base level” encryption, still the lowest encryption level with the 40 bit long symmetric key.
  - The “SSL Level 2" encryption, for stronger exportable algorithms such as DES.
  - The “SSL Level 3" encryption, equivalent to the previous “domestic level”.



Cipher specifications supported by System SSL

- **SSL V2**
  - 1-RC4 US
  - 2-RC4 Export
  - 3-RC2 US
  - 4-RC2 Export
  - 6-DES US
  - 7-TDES US

  SSLBase = 42
  Security Level 2 = 642
  Security Level 3 = 764321

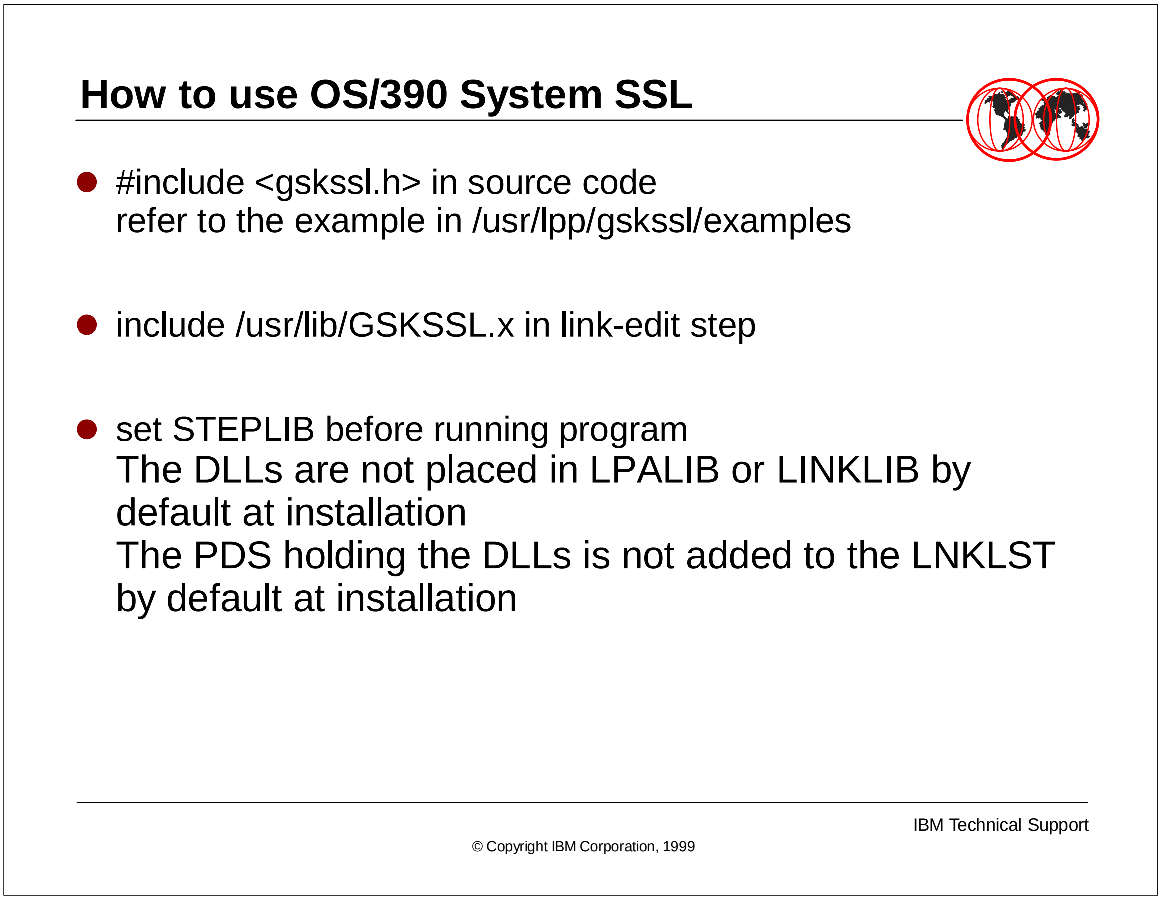
- **SSL V3**
  - 01-NULL MD5
  - 02-NULL SHA
  - 03-RC4 MD5 Export
  - 04-RC4 MD5 US
  - 05-RC4 SHA US
  - 06-RC2 MD5 Export
  - 09-DES SHA
  - 0A-TDES SHA US

  SSLBase = 03060201
  Security Level 2 = 0306090201
  Security Level 3 = 05040A0306090201

IBM Technical Support

© Copyright IBM Corporation, 1999

- Depending on the SSL version selected in the gsk_secure_soc_init() call, a suite of cipher algorithms is to be negotiated during the handshaking.
- The contents of this suite also depends on which fmids are installed in System SSL.



How to use OS/390 System SSL

- #include <gskssl.h> in source code
  refer to the example in /usr/lpp/gskssl/examples

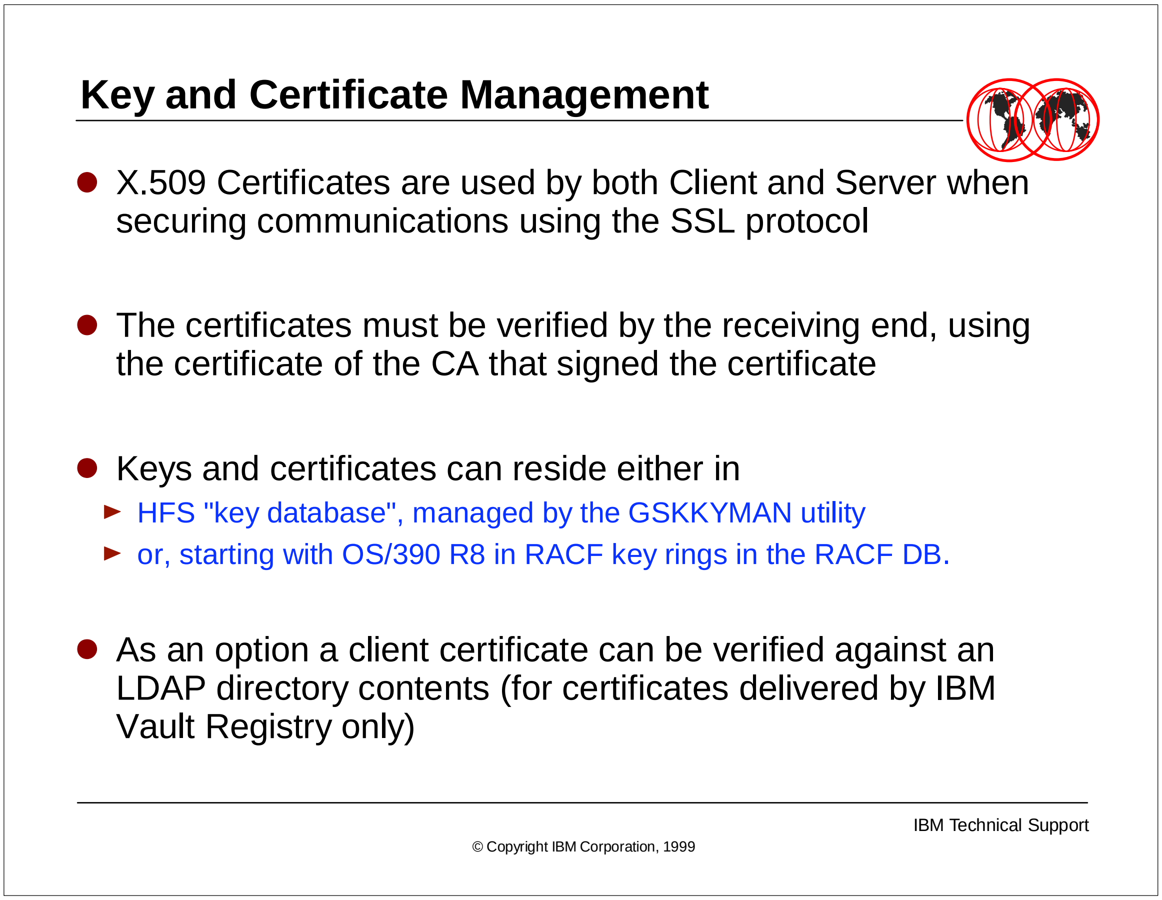
- include /usr/lib/GSKSSL.x in link-edit step

- set STEPLIB before running program
  The DLLs are not placed in LPALIB or LINKLIB by default at installation
  The PDS holding the DLLs is not added to the LNKLST by default at installation

IBM Technical Support

© Copyright IBM Corporation, 1999

- The header file gskssl.h must be included in the source C/C++ code.
- The source program must be link-edited with the GSKSSL.x file.
- As the DLLs are shipped in PDS format, and as the library is not, by default at installation, in the LPALIB or LNKLST, insure that the OS/390 library is made known of the C/C++ applications using the STEPLIB environment variable.



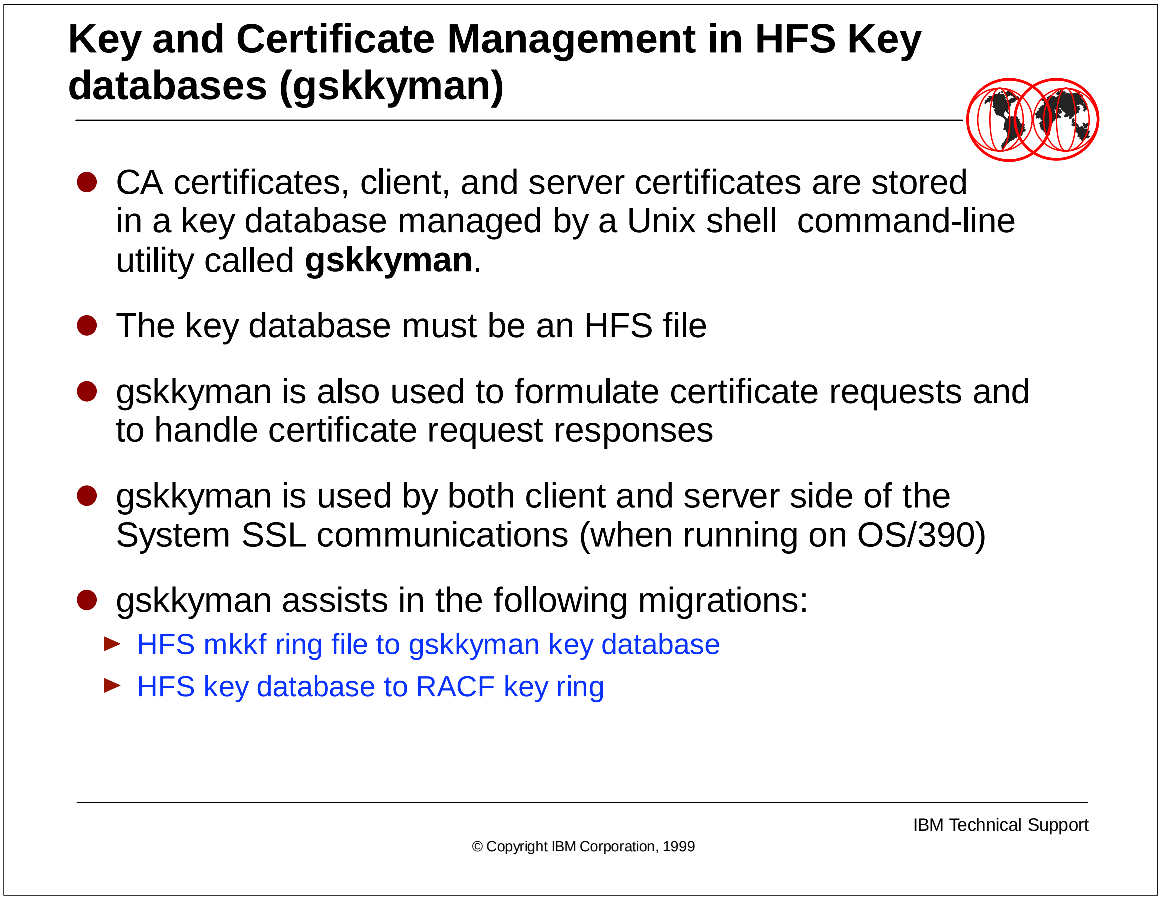
Key and Certificate Management

- X.509 Certificates are used by both Client and Server when securing communications using the SSL protocol
- The certificates must be verified by the receiving end, using the certificate of the CA that signed the certificate
- Keys and certificates can reside either in
  - HFS "key database", managed by the GSKKYMAN utility
  - or, starting with OS/390 R8 in RACF key rings in the RACF DB.
- As an option a client certificate can be verified against an LDAP directory contents (for certificates delivered by IBM Vault Registry only)

IBM Technical Support

© Copyright IBM Corporation, 1999

- X.509 certificates are the vehicle to transfer public key and authentication data during the SSL handshake phase between the client and the server.
- Each side must have the proper certification authority certificate(s) installed and accessible in order to validate the certificate transmitted by the other side.
- The side’s certificate(s) along with the proper certification authority certficate(s) can be kept in HFS files and managed by the gskkyman utility, or, beginning with OS/390 R8, can be kept in key rings in the RACF DB.
- Specifications can be given to system SSL in a server end, to access an LDAP directory contents to verify a client certificate.



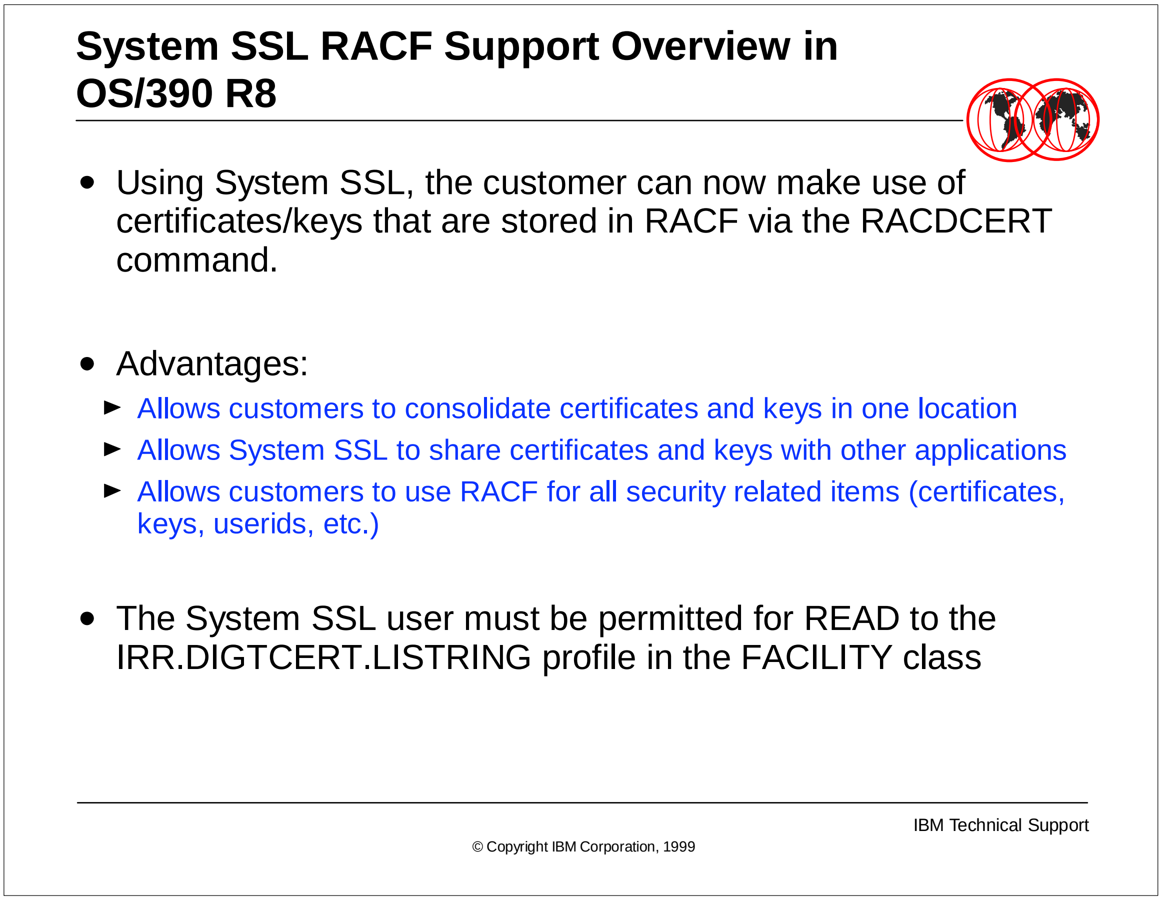
Key and Certificate Management in HFS Key databases (gskkyman)

- CA certificates, client, and server certificates are stored in a key database managed by a Unix shell command-line utility called **gskkyman**.
- The key database must be an HFS file
- gskkyman is also used to formulate certificate requests and to handle certificate request responses
- gskkyman is used by both client and server side of the System SSL communications (when running on OS/390)
- gskkyman assists in the following migrations:
  - HFS mkkf ring file to gskkyman key database
  - HFS key database to RACF key ring

IBM Technical Support

© Copyright IBM Corporation, 1999

- gskkyman is delivered with System SSL and allows to create and manage certificates and keys in HFS key databases
- it is a replacement for the mkkf facility, delivered with OS/390 LDAP and web servers prior to OS/390 R7. mkkf is used to create and manage keys and certificates in HFS key rings.
- gskkyman provides the necessary functions to migrate from HFS key rings (mkkf based) to HFS key databases (gskyman based), and to migrate from key databases (gskkyman based) to RACF key rings



System SSL RACF Support Overview in OS/390 R8

- Using System SSL, the customer can now make use of certificates/keys that are stored in RACF via the RACDCERT command.
- Advantages:
  - Allows customers to consolidate certificates and keys in one location
  - Allows System SSL to share certificates and keys with other applications
  - Allows customers to use RACF for all security related items (certificates, keys, userids, etc.)
- The System SSL user must be permitted for READ to the IRR.DIGTCERT.LISTRING profile in the FACILITY class

© Copyright IBM Corporation, 1999

- As of OS/390 R8, RACF can be used to create keys and certificates and to manage and store them in the RACF DB, using the RACDCERT GENCERT command.
- The certificates are grouped by "key rings" in the RACF DB. Key rings are created by RACDCERT ADDRING comand, and certificates are connected to a key ring using the RACDCERT CONNECT command.
- The intent is to provide the user with a single location where all security objects and policies can be stored and managed. The RACF DB is used for this purpose, with the advantages inherent to its location within the system and its historical background.

Publications

- System SSL related materials:
  - OS/390 Security Server (RACF) Command Language Reference - SC28-1919
    - http://w3.enterlib.ibm.com/cgi-bin/bookmgr/BOOKS/ICH1A405/CCONTENTS
  - Cryptographic Services System Secure Socket Layer Programming Guide and Reference - SC24-5877
    - http://w3.enterlib.ibm.com/cgi-bin/bookmgr/BOOKS/GSKSSL00/CCONTENTS
  - Additional information on the SSL protocol can be found at:
    - http://sitesearch.netscape.com/newsref/ssl/3-SPEC.html
    - http://info.internet.isi.edu:80/in-drafts/files/draft-ietf-tls-protocol-05.txt

© Copyright IBM Corporation, 1999