CSM-1000-0

IBM Crypto Server Management

General Information Manual

Notices

The functions described in this document are IBM property, and can only be used, if they are a part of an agreement with IBM.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectable rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

First Edition, May 2000

This edition, CSM-1000-0 applies to DKMS (Distributed Key Management System) Key Management Workstation Release 5.7.00 (DKMS5700) and DKMS Key Management/MVS Release 3.10 (KMG0310P.*) and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality.

Comments may be addressed to your IBM representative or the IBM branch office serving your locality. IBM may use or distribute any of the information you supply in any way considered to be appropriate without incurring any obligation to you.

Contents

Overview

Summary

Crypto Server Management (CSM) makes it possible to centralize the management of IBM 4758 PCI Cryptographic Coprocessors used in remote computers.
CSM makes it possible to bring the administration of the 4758 Coprocessors from the remote location to one central location, from which all 4758 code and keys can be controlled.

The business benefits of this tool are:

- minimizing the need for local crypto-skilled personnel
- minimizing on-site support of crypto coprocessors from skilled personnel
- enabling a quick recovery after an unplanned stop of crypto coprocessors
- new crypto-functions, and new keys can be introduced centrally with no need for local procedures
- no need for unsecured or difficult procedures for backup of keys
- no need for shipping initialized 4758 coprocessors (the tamper resistance of a 4758 makes it sensitive for physical handling with the risk that a initialized 4758 is useless at arrival and must be reinitialized)

The security benefits are:

- no vulnerability of local keys
- no exposure of exchanged keys
- central control over code and all keys in the network
- defined level of security can be easily enhanced when needed
- no need for having tight control over shipped initialized 4758s.

Crypto Server Management Introduction

Crypto Server Management makes it possible to manage the 4758 Coprocessor in remote computers from a central location.

It is the idea behind CSM that dedicated personnel shall concentrate on their mission rather than spend their time on technical work.
The technical work is removed to a central location, typically a DP center, where technical knowledge is present and from where all the cryptographic devices and keys in the remote computers are administered in any kind of network.
Typically, the far end nodes of a network will have a computer serving as a Crypto Server for a number of clients (hence the term Crypto Server). This is done in order to ensure authentication, confidentiality, and integrity between the Crypto Server and the Central Computer (also a kind of Crypto Server) in the middle of the network - or between the Crypto Servers themselves.

These Crypto Servers can be situated in a branch, in an office or the like, where the necessary knowledge on how to operate the Crypto processors are not present.

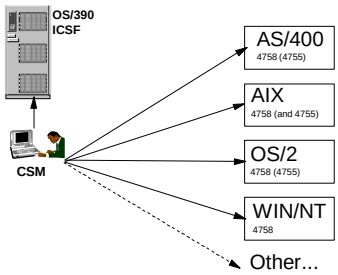
Crypto Servers could also be situated in machine halls, in a Remote Backup center, at a Business Partner, or the like where the knowledge might be present, but where either the time it takes to initialize or a difficult set-up of manual procedures makes it attractive to centralize the management of such Crypto Servers.
In short, CSM solves the problem of being force to maintain local technical knowledge and makes it possible to centralize all management of Crypto Servers.
This implies that also the problem of exchanging keys securely between the Crypto Servers or between the central location and the Crypto Servers is solved by CSM - without introduction of challenging or unsecure local manual procedures.

Even if CSM is concentrated around IBM 4758, it can also support the IBM 4755 Crypto adapter, however, only the key exchange part of CSM. See appendix A for details on IBM 4755 support.

Crypto Servers Set-up

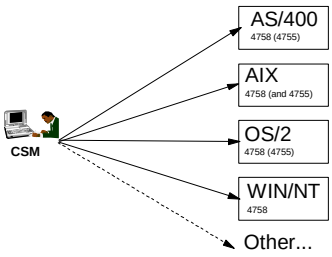
From a CSM point of view, a set-up of Crypto Servers in a network can be of any kind.
It can support e.g. a Branch network where a lot of branches each have one or more Crypto servers.
Or, it can support a more centralized set-up where the Crypto Servers to manage all are centrally located in one or more machine halls.
Both set-ups can be separately defined within CSM, but it can also be mixed, making it possible that keys at the central Crypto Servers can also be present at the Branch Crypto Servers.
How a specific Crypto Server set-up looks at each customer, or how different Crypto Server set-ups are mixed is a fully customizable possibility.

A Crypto Server set-up might look as illustrated below, if a central host communicating with the Crypto Servers is involved. This is called the CSM On-line version:



The On-line version enables keys generated for the Crypto Server to be directly available also for OS/390 ICSF (or IBM 4753) on up to 30 LPARs, making it possible to ensure communication between host and Crypto Servers.
All keys are stored on OS/390 DB2 tables, taking advantage of normal back-up and recovery procedures on OS/390.

If the set-up does not include a central host, CSM in the Stand-alone version still makes it possible to administer the Crypto Servers:

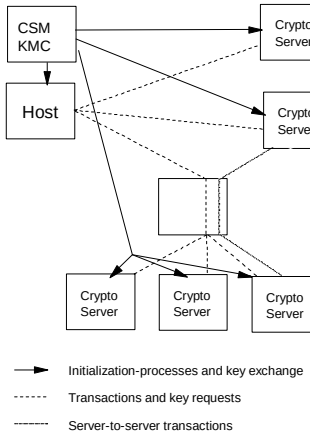


In the Stand-alone version, keys are stored locally on the CSM Workstation (and must be backed up locally).

As indicated by “Other...” in both drawings, the CSM has no problem supporting exchange of keys with other platforms and other Crypto engines than the IBM 4758 and 4755.
CSM can be customized to support other Crypto Architectures than CCA (Common Cryptographic Architecture) from IBM.
From the Key exchange perspective, it is therefore possible to generate and exchange keys with non-IBM Crypto engines. It is also possible to mix the hierarchy so that keys are generated for both the IBM world and other Crypto engines.
From CSM perspective, all that needs to be done is to define the new Crypto architecture within CSM and exchange the 4758/4755-APIs of the CSM programs on the Crypto Server with the APIs from the relevant Crypto vendor.

Crypto Server Management Topology

The design of the CSM in its full version is based on the assumption that a DP centre with a central host owns a network with Crypto Servers having a 4758 installed and that these are all managed from a central system:

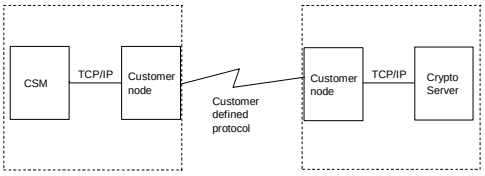


The Crypto Server is a Computer running AS/400, AIX, NT or OS/2 with a 4758 coprocessor installed. The server can be separate or part of a shared pool through a front-end.

The CSM KMC (Key Management Centre) makes all initialization processes, holds information of all Crypto Servers and 4758 in the system, and controls all keys in the system.

The Host is the platform used by the customer for the transactions using cryptographic functions and for possible key requests from the server.

The communication protocol between CSM KMC and the Crypto Server is based on TCP/IP. But it is possible for the customer to use another protocol in the actual network. This only requires introduction of protocol converters between CSM KMC and the Crypto Server(s).
The nodes doing protocol conversion can be placed on the same physical machine as the CSM KMC and/or the Crypto Server or on a separate machine.
However, the communication between both CSM and the protocol converter and the protocol converter in the other end and the Crypto Server will still be using TCP/IP. So the protocol converter program will be a matter of converting TCP/IP to whatever protocol is used and back again:



Crypto Server Security

Since the communication between CSM and the Crypto Server will be based on a physical network, it is obviously of the utmost importance that the authentication of both CSM and each individual Crypto Server can be trusted and that communication between CSM and the Crypto Server can be secret.

As an integrated part of the communication between CSM KMC and the Crypto Servers, CSM offers a sophisticated variety of security methods to protect this communication. The protocol between CSM and a Crypto Server uses:

- Digital Signatures from both CSM and the Crypto Server using 1024 bit RSA keys
- Key Exchange using a mixture of 1024 bit RSA keys and 112 bit DES keys
- Hashing of vital information using SHA-1.

Furthermore, only approved 4758 code (so-called segments) can be loaded into the 4758 Coprocessor - as an integrated part of the 4758 load process.

The last item is the confidentiality of the master key of the 4758. This key protects all other keys in the 4758 Key Storage, so it is of vital importance that this key cannot be exposed. CSM makes it possible always to have the master key generated and kept within the 4758. Recovery of keys in case of errors is simply changed from dependency on the Master key to dependency on the set-up of the key hierarchy.
Also, we provide guidelines on how to secure the 4758 Master-key within the 4758 concept.

IBM 4758

IBM 4758 Introduction

IBM 4758 PCI Cryptographic Coprocessor is a secure, tamper resistant crypto coprocessor which is FIPS 140-1 level 4 certified. It is designed with the purpose of being applicable in a hostile environment and can therefore be used in environments where the security relies on the coprocessor itself and not on the surroundings. Thus, it is possible to use an IBM 4758 Coprocessor in environments where actual knowledge of how to maintain it is not present.

Consequently, it is possible to administer any 4758 from a central place without interacting with anybody at the 4758. Physical intervention is then only needed when installing the 4758 coprocessor itself, its drivers, plus the administration tool. However, since it is a normal PCI/"plug and play" component, this is quite easy.

This means that a DP center, administering a network of any kind, will be able to install a 4758 in all their cryptographic nodes, and then - from a central location - administer all other processes needed in order to have the coprocessor working without physical interaction with the 4758 itself.

Security Challenges administering 4758 in a Network

When administering a 4758 in a remote server from a central location, some security challenges must be solved. This is done by CSM.
Besides trusting the Crypto coprocessor itself, the security challenges are to ensure that the owner or administrator of the network system can control all the cryptographic nodes in this system.

This means that the owner of the network must have a system to centrally define:

All cryptographic nodes in his network

A spoofing cryptographic node will be able to place itself in the middle of the network, possible with the ability to receive keys, pins etc. from other nodes!

All code to be loaded into the Crypto coprocessor

If we cannot control the code, we do not control the functions of the Crypto coprocessor - perhaps it is able to change the intended use of a key by allowing it to come out in the clear!

All keys in all cryptographic nodes

If the keys cannot be controlled, a key can be used for other purposes than intended, e.g. a key meant for encrypting other keys can be used for deciphering data, thereby enabling encrypted keys to be deciphered!

As for the keys, it is - as always - the exchange of the first key that causes the biggest problem. Using a public key scheme for exchanging keys, however, makes it a matter of being sure that keys are exchanged with a known node only.

Lastly, it is important to state that keys used for key management purposes should be unique for each Crypto Server in order to minimize the damage in case of a compromised key in a certain server.

The combination of 4758 coprocessor as the trusted Cryptographic coprocessor and CSM as the Central Management tool makes it possible to solve these challenges, and thereby to have trusted Crypto Servers in the network.

Initialization Process of a 4758

Initialization of a 4758 can be logically separated into three parts:

Loading of code
Load 4758 Segment one to three

Initialization of the 4758
Load Function Control Vector (FCV). Set default roles and miscellaneous parameters. Initialize key storage(s) and generate a random Master key.

Exchange of keys
Exchange keys with the IBM 4758 and write to Key Storage(s).

The first two parts of the process are normally those referred to in connection with 4758 initialization. These two tasks are more or less the same for all customers and Crypto Servers. However, the definition of the Default role needs to be seen in conjunction with the keys and functions used on the remote cryptographic node.

The last process - exchange of keys - is individual for each customer and Crypto Server, depending on the keys and functions used on the cryptographic node.

Crypto Server

Prerequisites

Prior to the initialization of the 4758, the 4758 itself must be physically installed at the server, and furthermore the software needed to access the 4758 must be configured, and the device drivers must be started.
The 4758 Segments and FCV must be installed and loaded in order to verify that the 4758 works after installation.
This is the only task to be performed locally at the Crypto Server - as described in the 4758 Installation manual, and it has to be done only at the first time installation or if the 4758 fails in such a way that it must be replaced.
Optionally, personal identification and/or a password can be entered as part of the physical installation for a later comparison with the same kind of data stored at the central location.

The Crypto Server part of the CSM software plus a file holding the Public key from the CSM KMC must be installed on the Crypto Server.
This will probably take place as part of the normal procedure when installing own written business applications on the Crypto Server.

The CSM software on the Crypto Server must be started and will act on data sent from CSM KMC and use these data to initialize the 4758, receive the exchanged keys, and install them in 4758 Key Storage. Only data from a proper CSM KMC will be accepted.

Functions

The process of initializing the 4758 on the Crypto Server from the central location includes:

- Loading of segment 1, 2, and 3
- Function Control Vector loading
- Setting of the default role
- Initialization of the DES key storage file
- Initialization of the PKA key storage file
- Setting misc. parameters e.g. date/time, environment ID
- Generation of a random master key.

When these steps have been carried out, the 4758 is in a running state.
Using CSM, the operator can then perform the following from the central location and at any give time:

- Show a detailed status of the IBM 4758 coprocessor
- Exchange one or more Key-Encrypting Keys with the CSM KMC using a PKA scheme for the first exchanged key
- Install, delete, list, and test key(s) in key storage(s)
- Get/send files from/to Crypto Server
- Reinitialize the 4758, i.e. repeat the above initialization steps - only if the 4758 is idle.

CSM Key Management Centre

Prerequisites

CSM SW and HW must be installed and attached to the host (On-line version). See Appendix C, Components, for details.

Functions

CSM is able to manage a large number of Crypto Servers.

Each Crypto server can be managed independently (own key hierarchy, own settings), but it is also possible to manage a group of remote servers with shared key hierarchy and the same settings.

It is possible to define that only parts of the initialization process and/or key exchange shall be performed.

All customization of Crypto Servers, 4758-related information, keys etc. can be done beforehand. This means that all necessary information is present at the time of the actual initialization process or part of it.

All actions within Crypto Server Management can be access controlled. The users defined in Access Control are verified against the User block in the Chip cards that are used when logging on to the CSM KMC.
CSM comes with a GUI interface in which the proper set-up of authorization for each action can be defined.

Also an Audit function is integrated in the Crypto Server Management. This means that all actions performed are logged and stored in a DB2 UDB table and/or - in the On-line version - as SMF records. A tool for viewing the Audit DB2 UDB table is also included.

CSM includes screens and functionality to:

- maintain information related to the Crypto Server - create, read, update, and delete the Crypto Server information
- use name, address etc. from the customer's own Crypto Server Database - in the On-line version - or update this information within the CSM KMC
- register the 4758 segments, function control vector and roles
- build a 4758 role
- attach a specific layout of a 4758 to a specific Crypto Server
- define a Crypto Server to be part of a pool of servers
- specify and define the keys to be exchanged with the server (see next chapter)
- query the status of the Crypto Server
- query the keys of the Crypto Server for presence and validity
- set default parameters used by these functions
- generate a CSM private/public key pair

register and run a 4755 load program, if 4755 is used (see Appendix A).

Besides these functions, the central part of communicating with the Crypto Server - including commands towards the server - is part of CSM KMC.

Keys

Crypto Server Key Hierarchy

For each group of servers a separate key hierarchy is defined. The key hierarchy is defined beforehand and can be a combination of specific Crypto Server keys and shared keys among a group of Servers. The keys can be generated beforehand or as part of the initialization process. The counterpart of each key can be installed at the central location on OS/390 ICSF or 4753 - and/or exchanged with other cryptographic nodes of the network.

The keys are defined within the scope of the Key Definition Table (KDT). KDT is a central DB2/UDB table holding all key-related information but not the keys themselves. Here you define the attributes of the key, such as key type, key ID, exchange methods, where to store it, etc. This table makes it possible to define whatever keys are needed on the specific system and in the appropriate number of versions.
CSM comes with a GUI interface managing all information in KDT.

Server Recovery Key

If the Crypto Server itself generates or receives keys from another node than the CSM KMC, these keys are not managed by the CSM.
In such a case, IBM can provide a method of bringing in the keys under the control of a Server Recovery Key using the CSM key hierarchy.

CSM Roles

How you are managing Crypto Servers in your network varies, depending on the angle from which you see the system.
For instance, setting up the system is a different task than actually initializing the 4758 when maintaining a Crypto Server.
A system using Crypto Servers is also part of a security solution, so besides being manageable, such a system must also be secure.

Consequently, the procedure of managing Crypto Servers in the network is not just a straightforward process for one person, but a mixture of different responsibilities in the organization taking care of the system.
In the process of implementing such a system, one must define these different fields of responsibility and allocate the appropriate personnel to each field.

Using CSM, we can see the need for four different fields of responsibility. These are:

Server System designer/architect
This person adapts the CSM to the designated system and level of security. This will only take place at the very beginning and if changes to the Server system are introduced

Key Manager
This person performs the necessary key management functions related to CSM, when the CSM Signature Key has to be generated or when non-server specific keys have to be generated. This will only take place at the very beginning or if new non-server specific keys are introduced or changed.

Server Administrator
This person identifies a specific server in the network for CSM. This will take place

every time a new Crypto Server is introduced in the network or if the information connected to the server changes (TCP/IP-address, physical address etc.).

Server Operator
This role performs the actual initialization of the 4758 and the key exchange with the server. This will take place whenever a new server is introduced in the Network, if new keys shall be exchanged with the server, or when the 4758 is reinitialized due to a failure. Furthermore, this role must be able to perform level-1 support of the Crypto Servers (using CSM).

Each role has a specific task to take care of within the CSM. The final implementation of these roles and tasks will naturally vary, depending on actual needs as well as on customer organization.
Possibly, the Key Manager will also generate keys for a specific Crypto Server.
Furthermore, one person or group of persons might assume more than one role.

Appendix A - IBM 4755 Initialization

It will be possible to administer a Crypto Server with a 4755 installed from the CSM KMC. However, some differences between the 4758 and the 4755 make the approach different - both regarding practicalities and security aspects.

Regarding practicalities, we must distinguish between the three parts of an initialization process for the 4758, defined in the introduction:

Loading of segments

Initialization of the coprocessor

Exchange the keys

The first process - loading of segments - is not part of an initialization process of a 4755. An equivalent process on a 4755 is downloading of micro-code into the 4755, which requires physical presence at the 4755. Consequently, this is not part of the CSM-initialization process, nor can it be.

The second process - initialization of the coprocessor - does have its equivalence on a 4755 and can be performed using CSM.
Initializing a 4755, however, is performed differently using a so-called Batch Load File or using a program. Thus, the program actually initializing the 4755 can be the program already doing it, or it can be coded by IBM on request.
The program can be executed on the Crypto Server through CSM.

The third process is the same, whether the receiving coprocessor is a 4758 or a 4755. Therefore, this part of the process will be fully supported by CSM, however, the PKA92 part of a 4755 must be initialized and loaded. If not, a new version must be obtained or new micro-code must be loaded. .

Regarding security, two concerns must be noted:

generating a master key on a 4755 actually reveals the master key in storage during the process.

the 4755 does not render the same services in order to verify code within the coprocessor.

Appendix B - Road Map

Authentication of the Crypto Server

In this version, CSM uses a manual procedure when initializing the 4758 for the first time in order to guarantee the authenticity of the 4758.
CSM includes a possibility for storing of personal information on the central site, (e.g. name, employee-number) of the person installing the 4758 as well as a password to be used when installing etc. This information can be communicated using other means than the network, e.g. via the phone, and it will be verified at the time of initialization.
It is customizable whether these features are going to be used or not.

It is the intention that future CSM will support a fully automatic authentication of each 4758 using asymmetric cryptographic methods.

RSA Keys

Today many customers want to have private RSA keys only available for the Crypto Server on which they are generated. Thus there is no need for storing such keys on the CSM KMC. However, if customers want that ability, CSM can be extended to store asymmetric keys, e.g. for restoring purposes.
This means that in current version of CSM, asymmetric keys can be generated, stored, viewed, and deleted locally on the Crypto Server, but they cannot, as symmetric keys, be stored at the central location.

CSM KMC

In the current version, CSM KMC runs on OS/2 using an IBM 4755 Crypto adapter together with an IBM 4754 Security Interface Unit and with APPC connection to an OS/390-host.
It is our intention, within this year, to deliver CSM KMC a Windows NT application using IBM 4758 as Crypto adapter and TCP/IP connection to the host.
Furthermore, a replacement for IBM 4754 and PSC will be introduced into CSM, making it possible to use Smart Cards for log on to the 4758.

Appendix C - Components

Crypto Server

Crypto Hardware: IBM 4758 coprocessor model 001, 013, 023, and 002
(or IBM 4755 Crypto adapter)
Operating system: NT, AIX, AS/400, OS/2.
Communication software: TCP/IP.

CSM Workstation

Current version:
Crypto Hardware: IBM 4755 and IBM 4754
Operating system: OS/2
Database System: DB2 UDB for OS/2
Communication software towards host: APPC
Communication software towards Crypto Server: TCP/IP

Next version:
Crypto Hardware: IBM 4758 Coprocessor model 023 or 002
Operating system: Windows NT
Database System: DB2 UDB for NT
Communication software towards host: TCP/IP
Communication software towards Crypto Server: TCP/IP

Host

Crypto Hardware: CCF, PCI-CC and IBM 4753
Database System: DB2 UDB for OS/390
Programming language: COBOL for OS/390

Current version:
Communication with WS: APPC and CICS Transaction Server

Next version:
Communication with WS: TCP/IP and started task.

Customer Components

Optional use of a customer communication protocol between CSM KMC and the Crypto Server only requires coding of the protocol converter part that applies to the customer protocol.