ICSF Up and Running!

By Dan Little

SEVERAL years ago when I was asked to get the hardware cryptography on the mainframe up and running I was a bit lost to say the least. I'd been an MVS System Programmer for many years and heard about cryptography and had some vague notions about what it was but I was no cryptography expert. Where would I start? What do I need to know? In this article I intend to try to give good directions to get from zero to ICSF up and running.

ICSF (Integrated Cryptographic Service Facility) is software that ships as part of the z/OS Base element called z/OS Cryptographic Services. It works in conjunction with the cryptographic hardware on the mainframe. Programmers can call ICSF services to perform cryptographic operations ranging from generating random numbers, enciphering data, all the way through to performing PIN operations for ATMs (automated teller machines).

HARDWARE

In order to use ICSF you need to make sure you have the appropriate hardware in place or you won't be able to do much. The hardware on z990 and higher server consists of a feature called CPACF (CP assist for cryptographic processors) and one or more cards described below. The CPACF feature must be ordered to turn on the cryptographic functions built into each CP on the mainframe. It is also a prerequisite for using the cards below for secure cryptographic.

ICSF provides support for Cryptographic Coprocessors called PCIXCC and the new Crypto Express 2 CEX2C cards. They are PCI cards like on a personal computer but they are installed on a PCI bus in the z990 or the new z9-109 processor. These cards are secure tamper-proof enclosures. The only way to get any access to the functions of these cards is by using ICSF services.

If you want to provide secure cryptographic services like PIN processing for ATMs (automated teller machine) then you need to have at least one PCIXCC or CEX2C card installed and enabled.

FIGURE 1: USING PROGXX FOR LINKLIST

```
LNKLST DEFINE NAME(LNKLSTxx)
LNKLST ADD NAME(LNKLSTxx) DSNAME(SYS1.SCSFMOD0)
And adds for all your other linklist libraries
LNKLST ACTIVATE NAME(LNKLSTxx)
```

FIGURE 2: ADD THE LIBRARY TO PROGXX

```
APF ADD DSNAME(SYS1.SCSFMOD0) VOLUME(******)
```

Once you have the appropriate hardware installed and enabled you must work with your hardware person to assign a cryptographic domain to each LPAR. A domain is essentially a set of master key registers that are unique from the other 15 domains. A good practice is to assign one domain to each LPAR as its Usage Domain. The usage domain is the domain that this LPAR will use for its master keys. If you assign only one usage domain to the LPAR then you can omit the DOMAIN parameter in ICSF as it will use the one assigned to the LPAR. There are also control domains but this is only used for TKE (trusted key entry) which is a topic all on its own.

For more information about the hardware a good IBM Red book to reference is one called "IBM eServer zSeries 990 (z990) Cryptography Implementation" which you can find on the site http://www.redbooks.ibm.com.

INSTALLATION

When you order z/OS you get a version of ICSF with it. Depending on what level of z/OS you are running you may want to go to the web site

http://www-03.ibm.com/servers/eserver/zseries/zos/downloads/

and find the latest Web Deliverable and install it so you are starting with the most up-to-date version of the product that will run on your

z/OS release. If you do download a new version, follow the instructions provided to install the code before you proceed below.

Once the ICSF code is installed and available on your system you need to set up some MVS parameters, RACF, started task procedure for ICSF, parameters, and VSAM clusters.

MVS PARAMETERS

In order to run ICSF you need to add SYS1.SCSFMOD0 to the MVS linklist and if you are using PROGxx for linklist it would look something like FIGURE 1.

The library also needs to be APF (authorized program facility) authorized so you must add the library to PROGxx with the code in FIGURE 2.

ICSF has some authorized TSO commands and this requires an update to SYS1.PARMLIB(IKJTSO00). You must add both CSFDAUTH and CSFDPKDS commands to each of the AUTHCMD and AUTHTSF tables in IKJTSO00.

RACF

Decide what you want to call the ICSF started task. It can be anything you like but let's say you decide to call it ICSF. You need a started task userid which you would set up using the RACF commands. See FIGURE 3.

The ADDUSER defines a protected userid for ICSF since it is to be used by the started task only and makes the profile owned by group SYS1 and gives the user the default group of SYS1. Your installation may require you to use a different DFLTGRP and OWNER but the commands are the same so just fill in the blanks.

The RDEFINE STARTED is to tell RACF what userid and group to assign to the started task ICSF when a START ICSF command is issued.

The SETROPTS refreshes the profiles for STARTED class which is normally RACLISTed (i.e. profiles cached in a dataspace).

You should also investigate activating the CSFSERV and CSFKEYS classes in RACF as these are required if you want any control over what services can be used and what keys different users can access. Most likely you do not want a vital ATM key to be usable by just anyone on the system so this is quite important.

PROC

You need to set up a member called ICSF (or whatever you are calling your ICSF task) in SYS1.PROCLIB on the system where you want to start ICSF. IBM provides a sample procedure which you copy and rename to ICSF. The provided member is SYS1.SAMPLIB(CSF). As provided it expects your ICSF parameters member will be SYS1.PARMLIB(CSFPRM00). If that is what you want then you are done with the proc setup. If not change the proc so that the CSFPARM DD points to where your parameters will live. The dataset needs to be RECFM=FB, LRECL=80.

PARAMETERS

FIGURE 3: CREATE A STARTED TASK USERID

```
ADDUSER ICSF NOPASSWORD NAME('ICSF Task') DFLTGRP(SYS1) OW(SYS1)
RDEFINE STARTED ICSF.** STDATA(USER(ICSF) GROUP(SYS1))
SETROPTS RACLIST(STARTED) REFRESH
```

FIGURE 4: ERROR MESSAGE

```
CSFM101E PKA KEY DATA SET, CSF.SCSFPKDS IS NOT INITIALIZED.
CSFM419E INCORRECT MASTER KEY (BOTH) ON PCI X CRYPTOGRAPHIC
COPROCESSOR
X00, SERIAL NUMBER 93001234.
CSFM100E CRYPTOGRAPHIC KEY DATA SET, CSF.SCSFCKDS IS NOT
INITIALIZED.
CSFM001I ICSF INITIALIZATION COMPLETE
CSFM400I CRYPTOGRAPHY - SERVICES ARE NOW AVAILABLE.
```

The parameters for ICSF live in SYS1.PARMLIB(CSFPRM00) and there are sample members in SYS1.SAMPLIB called CSFPRM00.

To get started copy the sample member to your parameter library and make the changes required as we go along.

1. CKDSN—dataset name for a VSAM key-sequenced data set dataset called the CKDS (cryptographic key data set)—this is where your symmetric cryptography (DES(data encryption standard) and Triple-DES) keys will be stored for later use.
2. PKDSN—dataset name for a VSAM key-sequenced data set called the PKDS (PKA key data set) where your private keys will be stored. PKA stands for public-key architecture and are the type of keys used in certificates and other asymmetric algorithms.
3. DOMAIN(xx)—if you have assigned one and only one usage domain to the LPAR (logical partition) that will be running ICSF, then you can omit this parameter and the domain will be detected automatically. If the LPAR has multiple usage domains then you must select one and code the value in this parameter.
4. SSM (special secure mode) must be set to YES if you want to use the KGUP (key generation utility program) to generate keys or import keys. Otherwise you can set it to NO.
5. COMPAT(NO) should be coded if at all possible. COMPAT(YES) means that you want ICSF to provide the API for the ancient Programmed Cryptographic Facility software from IBM which allows you to use old applications using PCF without converting them to call ICSF. Be aware that although this is tempting it does impair your ability to make dynamic master key changes in ICSF as you must shut down ICSF. COMPAT(NO) allows seamless master key changes without disruption to the application.
6. COMPENC should be removed as it is no longer useful and is only tolerated by the software but doesn't do anything.

There are a number of other parameters but these are the critical ones for getting up and running.

CKDS AND PKDS DATA SETS

ICSF uses two datasets for permanent storage of keys.

- The CKDS is used to store DES keys in records that can be later referenced by label by programs using ICSF services.
- The PKDS is used to store public and private RSA (Rivest-Shamir-Adleman) keys used for (PKA) public key architecture processing. An example of PKA usage is digital certificates which can be generated using RACF RACDCERT commands with the ICSF option so that the private key of the certificate is stored in the ICSF PKDS encrypted under the RSA master key.

To define the CKDS use the JCL sample found in SYS1.SAMPLIB(CSFCKDS). All you need to do is define the dataset and leave it empty. When master keys are entered it will be initialized.

Similarly using SYS1.SAMPLIB(CSFPKDS) defines a PKDS for use with ICSF.

Once you have the datasets defined make sure that the ICSF started task userid has UPDATE access to the CKDS and PKDS datasets.

Take the dataset names used for the clusters and enter them in the CKDSN and PKDSN parameters in SYS1.PARMLIB(CSFPRM00) discussed above.

STARTING ICSF THE FIRST TIME

Once you have everything above in place you can start up the ICSF Started task by just issuing the command "START ICSF" (or whatever you called the proc) and it will come up but will give some error messages indicating there are incorrect master keys because you have not entered any master key yet. See FIGURE 4.

At this point there is not much you can do with ICSF as there are no master keys entered and so the PCIXCC cards will not be able to do much.

ENTERING MASTER KEYS

There are two master keys which you must enter. The first is the Symmetric or DES master key which is 32 hex digits per part. The parts are exclusive ORed together to come up with the final master key. Normally each of two or more people would enter a 32 hex digit key part known only to them and protected from the other key custodians.

The second key is the PKA (public key architecture) master key which has 48 hex digits per part. It is also entered by two or more people with parts known only to them so that no one knows the master key.

There are three ways to enter the master keys.

The first way which I have not used is Passphrase Initialization (PPINIT). The idea here is that you enter a passphrase which is a phrase which you must remember as you will need to know it when you need to re-enter the same master keys in the future. For example you could get additional PCI cards and have to load the same master key and if you don't know the passphrase you are up the creek without a paddle. The one advantage of this method is that for a first-time startup it will enter master keys easily and will also initialize your CKDS and PKDS which is required the first time you set up ICSF. The downside is that if you need to change your master keys at some point there is not an easy "passphrase init, reencipher, and change master key" panel. I would recommend using one of the two methods below instead.

FIGURE 5: OPTION 4 ADMINCNTL IN THE ICSF MAIN MENU

To change the status of a control, enter the appropriate character (E - ENABLE, D - DISABLE) and press ENTER.

```
Function                                 STATUS
--------                                 --------
.  Dynamic CKDS Access                   ENABLED
.  PKA Callable Services                 DISABLED
.  PKDS Read Access                      DISABLED
.  PKDS Write, Create, and Delete Access DISABLED
```

The second way is to use the ICSF ISPF panel dialog and enter the key parts there.

This method of key entry is described in the z/OS Cryptographic Services ICSF Administration Guide.

The other way which is considered the secure method is to also order a TKE (trusted key entry) workstation. This method encrypts and signs everything traveling from end to end between the workstation and the PCI cards. The master key is the most critical piece of information for cryptography as all other keys are ultimately based on the master key. If someone knows the DES (symmetric) master key then they can discover every other key in the cryptographic key hierarchy. Since it is so important one should consider using the TKE. The TKE method is described in the z/OS Cryptographic Services ICSF TKE Workstation User's Guide.

INITIALIZING CKDS AND PKDS

Once the master keys have been entered you will have to initialize the CKDS and the PKDS since they are empty.

For the CKDS, you use the ICSF panels to do this. Choose option 2 MASTER KEY, then option 1 INIT/REFRESH CKDS and then choose option 1 Initialize an empty CKDS and type in your CKDS dataset name. Hit Enter and you should see "INITIALIZATION COMPLETE" in the top right corner of the screen.

For the PKDS, you also use the ICSF panels. As before, choose option 2 MASTER KEY but then choose option 5 INITIALIZE PKDS and enter the PKDS dataset name. Hit Enter and you should see "INITIALIZATION COMPLETE".

ENABLING SERVICES

Since it is your first time starting up ICSF and getting the hardware initialized the initial settings of some options will be DISABLED.

To rectify this situation, you need to go to the ICSF main menu and choose option 4 ADMINCNTL. This will show you something like FIGURE 5.

You should enter an E beside each of the DISABLED items to activate all the services of ICSF.

As you hit ENTER you may see a bunch of messages on your screen talking about BOTH MASTER KEYS CORRECT ON X00 and so on. This is normal when you enable the PKA Callable Services.

STATUS CHECK

At this point, if you look in the system log you should be able to see a message saying "PKA Services Now Available" and the messages

around BOTH master keys being correct for each card that you have installed and entered master keys.

As a final check you should go into the main ICSF panel and choose option 1 COPROCESSOR MGMT and you should see your cards listed and they should all say ACTIVE. You can select them and scroll up and down to make sure the master key registers indicate VALID.

I hope this article has been useful to you and that it will help demystify ICSF a bit.

NaSPA member Dan Little has been an MVS System Programmer for 20 years. He has installed, supported and maintained numerous IBM and ISV products over the years. He works as a system programmer in Toronto, Ontario.