Using ICSF with IDCAMS REPRO

Introduction

Access Methods Services implementation for encryption is based on the older IBM cryptographic products. Since ICSF can support these older product macro calls, you can use IDCAMS REPRO to perform simple encryption and decryption. Certain limitations exist due to the older product support that do not allow the most robust, or secure use of cryptography. This paper does not address those issues but does attempt to provide you with a reasonable description of weaknesses and possible mitigating factors that may make the use of IDCAMS REPRO's cryptographic support acceptable in some situations. The purpose of this paper, however, is to provide a clear description of how to use IDCAMS REPRO ENCIPHER and DECIPHER options with ICSF and the cryptographic hardware.

Limitations

Keys

- Length of data encryption key is limited to 8-bytes or 56-bit DES. No Triple-DES support is available.
- Data key is either a clear value specification or a randomly generated value. My recommendation is that you NEVER use the REPRO parameter option of PRIVATEKEY. This exposes the clear data key value.

  The preferred method should be to use either EXTERNALKEY or INTERNALKEY with the parameter STOREDATAKEY also specified.

  EX/INTERNALKEY implies the use of a pre-established key-encrypting-key (KEK) value that already has been defined. An external KEK is one that will be used to encrypt the randomly generated data encrypting key for transmission to another OS/390 or z/OS system. That KEK's value must be defined on the receiving system for the REPRO DECIPHER to recover encrypted data key and thus, the data.

  An internal KEK is one that will be used to recover the encrypted key value that is expected to either be with the encrypted data or to be supplied as the value in the `SYSTEMDATAKEY` parameter on the DECIPHER statement. The internal KEK's value must be defined on the receiving system for the REPRO DECIPHER to recover the encrypted data key.

  STOREDATAKEY eliminates the need to keep up with the encrypted data key value. IDCAMS will handle storing the encrypted key with the encrypted key. You have the benefit of never losing the data key.
- Key Labels or the names you assign the key value are restricted to 8 characters due to the fixed size of IDCAMS REPRO storage areas. Although ICSF supports key labels of 64 bytes or 32 characters, use of longer length key names are not supported by IDCAMS REPRO's PCF implementation.

Cross-Platform Exchange

- Not possible without coding an application on the other platform and obtaining all the algorithm variables used by REPRO. Maybe with a lot of work you might be able to figure out the encryption algorithm variables used by IDCAMS REPRO. These are the variables that indicate which mode to use block or stream, the initialization vector, data length, etc.. Although this information is non-secret, IDCAMS REPRO does not externally document this information nor describe the 'packaging' of the final data. This lack of detail prevents the exchange of encrypted data with another system where applications could be written to decipher the DES algorithm provided the algorithm inputs used by REPRO were clearly defined.

Positive Points

- Key-Encrypting-Keys or the INTERNALKEY/EXTERNALKEY can be defined as a random 16-byte value rather than an 8-byte value. This is true if both the ENCIPHER and DECIPHER will be performed by ICSF and the crypto hardware. With a true 16-byte KEK, the 8-byte data key is Triple-DES protected making the encrypted data key extremely secure.

- While it is not possible to use Triple-DES data keys, one could ENCIPHER a data set requesting random number generation, then DECIPHER with a supplied known clear DATAKEYVALUE, and finally ENCIPHER again requesting a random number. While not true Triple-DES and very cumbersome, this as well as other techniques may be used to strengthen the protection of the data beyond encryption with a 56-bit (8-byte) DES key.

Usage Points

- ICSF must be running in compatibility mode, COMPAT(YES) specified in the ICSF Options Dataset. When operating in this mode, the hardware master keys cannot be changed dynamically, an IPL is required OR you can stop ICSF, change the parameter to COMPAT(NO), restart ICSF, do the master key change, stop ICSF, change the parameter back to COMPAT(YES), and restart ICSF again.

- If exchanging data with a PCF or CUSP system, this directions will cause problems. PCF and CUSP only support an 8-byte key length for the exporter and importer key values. The keys will need to be exchanged. If the key value is exchanged in the clear, no problem. If the key value is exchanged encrypted under another shared exporter/importer key, that key must be a NOCV key to remove the ICSF control vectors from the encrypted key being sent to the PCF or CUSP system or added to an encrypted key being received from one of those systems.

- For ICSF releases prior to z/OS V1R2 the data records to be encrypted must have a record length within the range of that specified by the ICSF MAXLEN Options parameter. If not, IDCAMS will return the following message in your JCL:

  ```
     IDC3336I  ** CIPHER RETURN CODE IS 148
  ```

  The total text length to be processed per call is affected by the REPRO CIPHERUNIT parameter. Thus, by specifying CIPHERUNIT greater than 1 you are in effect multiplying the length of text being sent for processing by the number of units(records) specified.

Using ICSF with IDCAMS REPRO

Step 1. Define the ENCIPHER INTERNAL or EXTERNAL key value or the DECIPHER SYSKEY key value.

1a. To exchange IDCAMS REPRO ENCIPHERed data with another OS/390 or z/OS system or to use for in-house data storage only you must define the key value to be used to encrypt and/or decrypt the data key. Key definition is done using ICSF key administrative options; KGUP or TKE. Examples of key definitions using KGUP are shown.

Table 1.

Condition		ICSF Key Type Requirement	Key Label Restrictions	Key Length Requirements	Other Special Requirements for ICSF Key Generation Utility
1. ICSF active to ICSF active	When ICSF system is the ENCRYPTer or SENDING party	EXPORTER	Limited to 8 characters	16 random bytes LENGTH of KEY=>16 or enter 32 digit key value supplied by partner	Clear Key => YES
	When ICSF system is the DECRYPTer or RECEIVING party	IMPORTER	Limited to 8 characters	16 random bytes LENGTH of KEY=>16 or enter 32 digit key value supplied by partner	
2. ICSF active to PCF active	When ICSF system is the ENCRYPTer or SENDING party	EXPORTER	Limited to 8 characters	16 random bytes LENGTH of KEY=>8 or 16 digit key value supplied by partner and entered as key value =>16 digits, 16 digits where the 16 digits are repeated	Clear Key => YES KEK must be a NOCV key, specify Control Vector ===> NO This implies that the ICSF CKDS must have NOCV keys defined. See the ICSF Administrator's Guide for more information.
	When ICSF system is the DECRYPTer or RECEIVING party	IMPORTER	Limited to 8 characters	16 random bytes LENGTH of KEY=>8 or 16 digit key value supplied by partner and entered as key value =>16 digits, 16 digits where the 16 digits are repeated	
3. ICSF in-house use: no exchange with another system		IMPORTER	Limited to 8 characters	16 random bytes LENGTH of KEY=>16	none

Examples of ICSF KGUP Key Control Statement Definitions

1. Define a shared Key-Encrypting-Key to use with ICSF PARTNER system using ICSF KGUP panels to create control statement for use in KGUP batch job.

```
---------- OS/390 ICSF - Create ADD, UPDATE, or DELETE Key Statement ----------
COMMAND ===>
Specify control statement information below

   Function ===> ADD___     ADD, UPDATE, or DELETE
   Key Type ===> EXPORTER    Outtype ===> ________    (Optional)
   Label ===> PRTNRKEK__________________________________________________
    Group Labels  ===> NO_   NO or YES
 or Range:
   Start ===> ________________________________________________________
   End   ===> ________________________________________________________

    Transport Key Label(s)
        ===> ________________________________________________________
        ===> ________________________________________________________
 or Clear Key              ===> yes       NO or YES

   Control Vector ===> YES  NO or YES    CDMF ===> NO_  NO or YES
   Length of Key  ===> 16_  8, 16 or 24  DES  ===> YES  NO or YES
   Key Values     ===> ________________ , ________________ , ________________
   Comment Line   ===> _______________________________________________________
```

Created control statement: ADD TYPE(EXPORTER) LENGTH(16),
CLEAR DES,
LAB(PRTNRKEK)

2. Define a shared Key-Encrypting-Key to use with PCF PARTNER system using ICSF KGUP panels to create control statement for use in KGUP batch job.

```
---------- OS/390 ICSF - Create ADD, UPDATE, or DELETE Key Statement ----------
COMMAND ===>
Specify control statement information below

   Function ===> ADD___     ADD, UPDATE, or DELETE
   Key Type ===> EXPORTER    Outtype ===> ________    (Optional)
   Label ===> PRTNRPCF________________________________________________________
    Group Labels  ===> NO_   NO or YES
 or Range:
   Start ===> ________________________________________________________________
   End   ===> ________________________________________________________________

    Transport Key Label(s)
        ===> ________________________________________________________________
        ===> ________________________________________________________________
 or Clear Key              ===> yes        NO or YES

   Control Vector ===> NO_  NO or YES    CDMF ===> NO_  NO or YES
   Length of Key  ===> 8__  8, 16 or 24  DES  ===> YES  NO or YES
   Key Values     ===> ________________ , ________________ , ________________
   Comment Line   ===> _______________________________________________________
```

Created control statement: ADD TYPE(EXPORTER) SINGLE NOCV,
CLEAR DES,
LAB(PRTNRPCF)

3. Define a Key-Encrypting-Key to use with in-house data using ICSF KGUP panels to create control statement for use in KGUP batch job.

```
---------- OS/390 ICSF - Create ADD, UPDATE, or DELETE Key Statement ----------
COMMAND ===>
Specify control statement information below

   Function ===> ADD___     ADD, UPDATE, or DELETE
   Key Type ===> IMPORTER    Outtype ===> ________    (Optional)
   Label ===> MFAREPRO________________________________________________________
    Group Labels  ===> NO_   NO or YES
 or Range:
   Start ===> ________________________________________________________________
   End   ===> ________________________________________________________________

    Transport Key Label(s)
        ===> ________________________________________________________________
        ===> ________________________________________________________________
 or Clear Key              ===> NO        NO or YES

   Control Vector ===> YES  NO or YES    CDMF ===> NO_  NO or YES
   Length of Key  ===> 16_  8, 16 or 24  DES  ===> YES  NO or YES
   Key Values     ===> ________________ , ________________ , ________________
   Comment Line   ===> _______________________________________________________
```

Created control statement: ADD TYPE(IMPORTER) LENGTH(16),
DES,
LAB(MFAREPRO)

1b. SUBMIT the JOB which produces a randomly generated key value for 16 bytes or 32 digits or a randomly generated single-length key value whose Left Half is the same as the Right Half to simulate a single length key. Single Length keys are used by PCF and CUSP. The Clear key Value will be in the ICSF KGUP CSFKEYS DD Data Set. Sample output contained in the CSFKEYS dataset for example 2 is shown below as three separate lines due to width restrictions but it is really a single line of output. This key value is shaded for ease of identification and displayed as HEX data also. The key value to be provided to the PCF partner system would be FE3E734A6254A2F7. This value is provided as a double-length key.

```
PRTNRKEK                                                        IMPORTERCLEAR
DDEDDDCD44444444444444444444444444444444444444444444444444444444CDDDDECDCDCCD444
79359252000000000000000000000000000000000000000000000000000000009476935933519000

                                                                ................
44444444444444444444444444444444444444444444444444444444444444440000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000

                ................Ú.Ë¢Âès7Ú.Ë¢Âès7................................
44444444444444440000000000000000F37465AFF37465AF00000000000000000000000000000000
00000000000000000000000000000000EE3A2427EE3A242700000000000000000000000000000000
```

1c. REFRESH the CKDS so that the key value can be used by REPRO.

Step 2. Ensure ICSF can supports PCF calls.

2a. To use IDCAMS REPRO ENCIPHER or DECIPHER with ICSF, the ICSF Option COMPAT(YES) must be specified in the ICSF Options data set at the start of the ICSF started task. If COMPAT(YES) is not specified, the REPRO job will fail with RC=12 and an error message in the JCL output of ** GENKEY RETURN CODE IS 4**.**

2b. Stop and Restart ICSF to have ICSF in COMPAT mode.

Step 3. Run the REPRO JCL.

EXAMPLES:

1. Encrypting a data set to send to another OS/390 or z/OS system having either ICSF or PCF active.

```
//ALLMONDE JOB  CLASS=A,MSGCLASS=O,NOTIFY=????????
//DEFINE EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD  SYSOUT=*
//SYSIN DD *
  REPRO                                   -
  INDATASET('ALLMOND.TEST(CLRTXT)')       -
  OUTDATASET('ALLMOND.TEST(OUTPUT2P)')       -
 ENCIPHER(EXTERNALKEYNAME(PRTNRKEK) STOREDATAKEY  CPHRUN(255))
//
```

Note: The keyname is not important as long as the key was specified with the parameters for the proper environment. Specifying the CIPHERUNIT parameter will enhance the performance of the encryption process.

Using ICSF with IDCAMS REPRO

To Decipher OUTPUT2P encrypted on and sent from another OS/390 or z/OS system.

```
//ALLMONDE JOB  CLASS=A,MSGCLASS=O,NOTIFY=????????
//DEFINE EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD  SYSOUT=*
//SYSIN DD *
  REPRO                              -
  INDATASET('ALLMOND.TEST(CLRTXT)')      -
  OUTDATASET('ALLMOND.TEST(OUTPUT2P)')      -
  DECIPHER(SYSKEY SYSKN(the key name on their system that represents the same value as PRTNRKEK))
//
```

2. Encrypting a data set to store securely on my OS/390 or z/OS system.

```
//ALLMONDE JOB  CLASS=A,MSGCLASS=O,NOTIFY=ALLMOND
//STEPD1 EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD  SYSOUT=*
//SYSIN DD *
  REPRO                              -
  INDATASET(ALLMOND.TEST.CLEAR)      -
  OUTDATASET(ALLMOND.TEST.CRYPTED)      -
  ENCIPHER(INTERNALKEYNAME(MFAREPRO) STOREDATAKEY CPHRUN(255))
//
```

Note: The keyname is not important as long as the key was specified with the Table 1 parameters for in-house use. Specifying the CIPHERUNIT parameter will enhance the performance of the encryption process.

To Decipher ALLMOND.TEST.CRYPTED and stored on my system.

```
//ALLMONDB JOB  CLASS=A,MSGCLASS=O
//STEPD2 EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD  SYSOUT=*
//SYSIN DD *
  REPRO   -
  INDATASET('ALLMOND.TEST.CRYPTED')      -
  OUTDATASET('ALLMOND.TEST.CLRDATA')      -
  DECIPHER(SYSTEMKEY SYSTEMKEYNAME(MFAREPRO))
//
```

REPRO Performance with IBM Hardware Crypto on Multiprise 3000 (2 CPs 1 CCF) and no other workload:

	Cipherunit of 1	Cipherunit of 255	approx %	Cipherunit of 1	Cipherunit of 255	approx %
ELAPSED TIME	00:00:02.11	00:00:00.57	56.49	00:01:43.86	00:00:24.26	76.64
CPU TIME(TCB)	00:00:01.50	00:00:00.14	87.27	00:01:20.21	00:00:006.54	91.85
(SRB)	00:00:00.01	00:00:00.01	-	00:00:00.27	00:00:00.26	-
SERVICE UNITS : CPU	37,960	3,555	90.63	1,988K	166,053	92.65
SERVICE UNITS : SRB	179	179	-	6,749	6,643	-
SERVICE UNITS : I/O	578	577	-	31,631	31,631	-
SERVICE UNITS : MSO	4,403	410	90.69	316,757	26,252	91.71
ACTIVE FOR	00:00:02.09	00:00:00.56	56.59	00:01:43.84	00:00:24.25	76.65
RESIDENT FOR	00:00:02.09	00:00:00.56	56.59	00:01:43.84	00:00:24.25	76.65
# OF RECORDS	17,922	17,922		1,011,006	1,011,006	