

Tales from de-crypt

How secure is your DB2 data?

BY JEFFREY BERGER AND JEFF NOVAK

In many industries encryption is a growing requirement to satisfy the need for data security. In some industries, government regulations, such as the Health Insurance Portability and Accountability Act of 1996, might actually impose a security requirement that is best met by encrypting data. A common misconception about DB2® and security concerns the value of encryption versus the traditional methods that DB2 uses to keep data secure from unauthorized usage. A combination of encryption and the more traditional methods of DB2 and RACF is the best defense.

DB2 and RACF work together to ensure that only authorized users can access DB2 data, but those security measures are ineffective against a person who can circumvent the operating system. For example, disposing of defective disks or obsolete control units can create a security exposure if the data on the disks is not encrypted. On the other hand, encryption itself is not a protection against somebody who illicitly gains access to a password because DB2 will happily decrypt data on behalf of an authorized user. Thus, encryption and userid/password controls are complementary aspects of security, helping to protect against different types of security exposures. In any case, it is always important to remember that dumps of the DBM1 address space be kept secure as it can contain information about encryption keys as well as unencrypted data in working storage.

Encryption methods for DB2

The technical challenges associated with encryption include application changes, performance overhead, and the difficulties of managing the encryption keys. The zSeries® platform provides the basis for meeting these challenges with the help of both hardware and software.

Consider these two fundamentally different methods to encrypt DB2 data on z/OS:

- The encryption method that requires Version 8 of DB2 for z/OS (column-level encryption)
- A separately purchased tool called the IBM Encryption Tool for IMS and DB2 Databases (row-level encryption).

There are advantages and disadvantages to either method. For example, the data encryption supported by DB2 Version 8 requires extensive application changes, and the encryption is done at the column level. This method also requires that the SQL users supply the encryption key. The chief advantage of DB2's column-level encryption over the tool is that the index is encrypted with the columns; however, DB2 does not support encryption of numeric columns, and the DB2 Load Utility does not support column-level encryption.

The IBM Encryption Tool for IMS and DB2 Databases (IET) uses row level encryption. This tool works on any version of DB2, and all DB2 utilities work with the encryption tool. The tool relies on the Integrated Cryptographic Service Facility (ICSF) to provide its centralized key management. An ICSF administrator manages the ICSF environment where the keys are built, stored and maintained. As a result, when you use this tool, DB2 applications do not need any awareness of keys.

Here's how it works. The tool uses DB2 Editprocs, and a key label is stored in the Editproc. The assignment of an Editproc to a table determines which key label to use for the table. ICSF itself determines the key and associates the master key with a key label as well as keeps track of these associations in its own CKDS data set. Thus, the security of the RACF-protected CKDS becomes the very important for encryption security.

Encryption and processor performance

Before the IBM @server™ zSeries 890 and 990 models, IBM processors supported "secure keys," which used the Cryptographic Coprocessor Facility (CCF). CCF has introduced the notion of secure coprocessor and master key. With a secure coprocessor, the application keys are kept outside the coprocessor encrypted under a master key. The only instance of the master key is inside the coprocessor. When invoking data encryption, the application passes to the secure coprocessor a copy of the application key to use, encrypted under the master key. The coprocessor decrypts the application key inside its physically secure enclosure and proceeds with data encryption. "Secure" in the phrase "secure coprocessor" means that nobody will see the actual value of the application key anywhere outside the coprocessor, whether it is in system storage or on a disk device.

Because CCFs were limited to CP 0 and CP 1, any application requiring encryption services needed to be dispatched on either CP 0 or 1. Thus, if many threads required encryption services, contention for these coprocessors could result in significant delays while the majority of the CPs remained underutilized.

CPACF, PCIXCC, and Crypto Express 2

The z990 processor introduced a new facility called CPACF, which does not carry the concept of master key and deals only with keys provided in clear ("clear keys"). This is why the IBM Encryption Tool for IMS and DB2 Databases uses the PCIXCC or Crypto Express2 coprocessors to secure the key with the master key, then use proprietary logic to invoke CPACF.



The PCIX Cryptographic Coprocessor (PCIXCC) and Crypto Express2 are new cards that provide sophisticated functions beyond those that the CCF provides. CPACF helps reduce both the CP time and elapsed time for encryption. CPACF consists of a new instruction, called KMC, which can execute on any CP. The KMC instruction uses “clear keys” instead of “secure keys” and is considerably faster than the older cryptographic instructions that CCF uses. Further, you don’t need to redispach an application on a different CP, and because the KMC instruction lends itself to direct use by database subsystems and access methods, KMC enables zSeries to reduce the software overhead of encryption. Bringing that instruction closer to the application is critical to minimizing the performance cost of encryption, especially with small rows.

Single DES and triple DES

ICSF and the IBM Encryption Tool for IMS and DB2 Databases support both single DES and Triple DES using clear key encryption. The performance cost of clear key encryption consists of two components. The first component is the overhead, which is largely a software cost, and the second component is a function of the data length, which can be called the hardware cost. Triple DES encrypts the data three times. It takes longer for a hacker to decode Triple DES than Single DES, measured in terms of years or decades, but keep in mind that the hardware performance cost of Triple DES is triple the cost of Single DES. Think of the choice between Single and Triple DES as a tradeoff between performance and security requirements. Security considerations should take into account the shelf life of the data. If the data becomes obsolete by the time that a hacker can break the code of Single DES, there is little or no value to using Triple DES.

Although the IBM Encryption Tool for IMS and DB2 Databases has existed for some time and supports secure keys, in order to use clear key functionality on the z890 or z990 processor, you must install PTF UK00049.

You also need to upgrade ICSF to FMID HCR770A, HCR770B or HCR7720, with one of the following PTFs:

PTF numbers and FMIDs:

- UA15677 HCR770A
- UA15678 HCR770B
- UA15679 HCR7720.

Hardware requirements for the IBM Encryption Tool for IMS and DB2 Databases

- A z890 or z990 server
- CP Assist for Cryptographic Functions (CPACF) DES/TDES Enablement (feature code 3863)
- Either PCIX Cryptographic Coprocessor (PCIXCC) (feature code 0868) or Crypto Express2 (feature code 0863).

Hardware requirements for CCF, CPACF, and PCIXCC or Crypto Express2

The following table summarizes the hardware requirements for CCF, CPACF, and PCIXCC or Crypto Express2:

Cryptographic Coprocessor Feature (CCF) (feature code 0800)	z800 and z900 models and lower
CP Assist for Cryptographic Functions (CPACF) DES/TDES Enablement (feature code 3863)	z890 and z990 models and higher
PCI X Cryptographic Coprocessor (PCIXCC) (feature code 0868)	z890 and z990 models and higher
Crypto Express2 (feature code 0863)	z890 and z990 models and higher

The following table summarizes the use of secure keys, clear keys, and DES levels on z800, z900, z890, or z990 processors that use CCF, CPACF, and PCIXCC or Crypto Express2:

Z800 or z900 and CCF	Z890 or z990 and PCIXCC or Crypto Express2	Z890 or z990 and CPACF
Single-DES (8-byte key) or Triple-DES (TDES 24-byte key)	Single-DES (8-byte key) or Triple-DES (TDES 24-byte key)	Single-DES (8-byte key) or Triple-DES (TDES 24-byte key)
Secure Key Encryption using ICSF services CSNBENC / CSNBDEC	Secure Key Encryption using ICSF services CSNBENC / CSNBDEC	Clear Key Encryption using ICSF services CSNBSYE/ CSNBSYD (IMS™ only) or KMC instruction (DB2 only) PCIXCC or Crypto Express2 feature is required because encryption keys are stored in CKDS and maintained by ICSF.

An example of performance overhead using IBM Encryption Tool (row-level encryption)

In most cases the performance of the IBM Encryption Tool is expected to be superior to DB2's column level encryption. (An exception is the case in which you only want to encrypt one or two small columns out of a very large row.) The performance cost can be characterized in terms of the overhead and hardware per-byte costs. The "overhead" includes the cost of invoking the Editproc plus the cost of encrypting at most 8 bytes. To compute the "per-row" cost, you would multiply the "per-byte" cost times the row length and add the overhead.

For example, on the z990 processor, the Triple DES "per-byte" cost is 0.0062 microseconds. The overhead of the IBM Encryption Tool is 0.84 microseconds per row. In addition, because the clear key architecture operates on chunks of 8-byte

blocks, another 0.22 microseconds of overhead exists if the row size is greater than 8 bytes but not a multiple of 8 bytes.

Consider a 164-byte row: you would estimate the total per-row CP cost of encryption to be $0.84 + 0.22 + 164 \times 0.0062 = 2.08$ microseconds. The overall effect of encryption on the CP time of an application will vary greatly depending on the complexity of the SQL. For an insert adding, two microseconds has the effect of adding 10% (if there are no indexes) or less to the CP time. On the other hand, for a "SELECT COUNT(*)" doing a table space, adding two microseconds per row may add 500% to the CP time. The relative cost of row processing typically lies somewhere between these two extremes.

The performance characteristics of the IBM Encryption Tool for IMS and DB2 Databases are similar to those of DB2 table space compression. In both cases, the performance impact is much less for online

transaction processing (OLTP) than for queries and utilities. The magnitude of CP cost is similar, and the row size can affect the cost.

The only disadvantage of the tool compared to DB2 column level encryption is that indexes are not encrypted. If it is essential that indexes be encrypted, you should probably choose another tool or security method.

Another problem is common to both the IBM Encryption Tool and DB2 column level encryption—namely that you cannot effectively compress encrypted data. Challenges remain for DB2 to merge the functional advantages for IMS DB2 Databases column level encryption and the IBM Encryption Tool for IMS and DB2 Databases, as well as to allow us to effectively compress encrypted tables. So stay tuned for more DB2 tales from de-crypt.

POP quiz for extra credit!

Identify RMF FICON channel types and generations

BY MARGARET PHILLIPS

Remember when all FICON® channels in an RMF™ channel path activity report looked very much alike? Well no longer. You might say that FICON channels have evolved into a few more types and generations than the last time you took this quiz.

The following figure shows a typical RMF channel path activity report. Can you correctly identify the two types of FICON channels shown? Four possible answers are provided below the figure. Choose the correct channel types and you win. "What do I win," you ask? Well, you win pride in being a channel type ace!

CHANNEL PATH ACTIVITY									
CHANNEL ID	PATH TYPE	G SHR	UTILIZATION (%)			READ (MB/SEC)		WRITE (MB/SEC)	
			PART	TOTAL	BUS	PART	TOTAL	PART	TOTAL
FE	FC	4 Y	4.83	9.02	0.80	2.40	2.69	0.80	0.88
FF	FC	1 Y	4.48	8.47	9.11	2.35	2.64	0.78	0.85

- a. FICON Express2 at 2 Gbit link speed
- b. FICON Express2 at 1 Gbit link speed
- c. FICON Express at 2 Gbit link speed
- d. FICON Express at 1 Gbit link speed

Answers: For the first line in the report, if you chose (a) to indicate that generation 4 was a FICON Express2 running with a 2 gigabits per second link speed, you are correct. If it had been generation 3, that would indicate a FICON Express2 channel auto-negotiated to 1 gigabit per second. For the second line in the report, if you chose (d) to indicate that generation 1 is a FICON Express channel auto-negotiated to 1 gigabit per second, you are correct. Had it been a generation 2, it would have been the same channel running with a 2 gigabits per second link speed. In summary, the Generation (G) field tells you a combination of which generation of which FICON channel is being used and the speed of the fibre link for this channel.