Craig S. Mullins

Return to Home Page

October / November 2007

The Resource for Users of IBM zSeries & S/390 Systems

z/JOURNAL

zData Perspectives

by Craig S. Mullins

DB2 Encryption Support Keeps Improving.

In this data-breach and regulation-laden world we currently find ourselves working in, the topic of data encryption has become increasingly popular. If we encrypt our sensitive data then only authorized folks who know the decryption key will be able to access it. And that significantly improves data protection.

So what can we do about encrypting sensitive DB2 data? Well, DB2 V9 offers some encryption news, but we need to go back a version to tell a complete story. You see, DB2 supports encryption in Version 8 through encryption functions that can be coded in to encrypt and decrypt DB2 data.

These functions allow you to encrypt and decrypt data at the column level. Because you can specify a different password for every row that you insert, you are encrypting data at the “cell” level in your tables. If you use these functions to encrypt your data, be sure to put some mechanism in place to manage the passwords that are used to encrypt the data. Without the password, there is absolutely no way to decrypt the data.

To assist you in remembering the password, you have an option to specify a hint (for the password) at the time you encrypt the data. The following SQL example shows an INSERT that encrypts a social security number using a password and a hint:

```
INSERT INTO EMP (SSN)
VALUES(ENCRYPT('289-46-8832',
               'TARZAN',
               '? AND JANE'));
```

The password is “TARZAN” and the hint we’ve chosen to provide is “? AND JANE”… so the hint will prompt us to think of Tarzan as the companion of Jane. In order to retrieve the encrypted data you will need to use the DECRYPT function supplying the correct password. This is shown in the following SELECT statement:

```
SELECT DECRYPT_BIT(SSN,'TARZAN') AS SSN
FROM EMP;
```

If we fail to supply a password, or the wrong password, the data is returned in an encrypted format that is unreadable.

The result of encrypting data using the ENCRYPT function is VARCHAR FOR BIT DATA. The encryption algorithm is an internal algorithm. For those who care to know, it uses Triple DES cipher block chaining with padding and the 128-bit secret key is derived from the password using an MD5 hash.

When defining columns to contain encrypted data the DBA must be involved because the data storage required is significantly different. The length of the column has to include the length of the non-encrypted data + 24 bytes + the number of bytes to the next 8 byte boundary + 32 bytes for the hint.

OK, that is the V8 stuff, but what about version 9? Well, DB2 9 for z/OS offers some nice improvements to encryption support. Firstly, DB2 can take advantage of encryption hardware advances.

CP Assist for Cryptographic Function (CPACF) is available on z990 hardware. CPACF can run on all the CPUs, but remember, this feature is available only on z990 and later machines, not the older z900. The z990 also introduces a PCIXCC card which is needed for the IBM Data Encryption Tool, but not for the DB2 encryption functions. The IBM Data Encryption Tool (available from IBM at an additional price) offers encryption for DB2 tables at the table level, whereas the encryption functions (free with DB2) offer encryption at the column level.

The CPACF delivers cryptographic support on every CP with Data Encryption Standard (DES), Triple DES (TDES), and Advanced Encryption Standard (AES)-128 bit data encryption/decryption, as well as Secure Hash Algorithm (SHA-1) and SHA-256 hashing. For a more detailed discussion of CPACF, associated technology and functionality, check out the IBM redbook: IBM eServer zSeries 990 (z990) Cryptography Implementation (SG24-7070).

Basically, the net result is that the cost of encrypting DB2 data under V9 is reduced on the z990 hardware. And IBM has added encryption support in the controllers of its storage devices, too.

OK, so far, we've been talking about encryption for data at rest. But DB2 9 for z/OS also improves support for encryption of data in transit. DB2 9 supports the Secure Socket Layer (SSL) protocol by implementing the z/OS Communications Server IP Application Transparent Transport Layer Security (AT-TLS) function. The z/OS V1R7 Communications Server for TCP/IP introduces the AT-TLS function in the TCP/IP stack for applications that require secure TCP/IP connections. AT-TLS performs transport layer security on behalf of the application, in this case DB2 for z/OS, by invoking the z/OS system SSL in the TCP layer of the TCP/IP stack. Support is provided for TLS V1.0, SSL V2.0, and SSL V3.0 protocols.

So encryption of data over the wire is improved in z/OS 1.7. The Communications Server supports AT-TLS, which uses SSL data encryption. SSL encryption has been available on z/OS for a long time, but now DB2 9 for z/OS makes use of this facility and offers SSL encryption using a new secure port. When acting as a requester, DB2 for z/OS can request a connection using the secure port of another DB2 subsystem. When acting as a server, and from within a trusted context, SSL encryption can be required for the connection.

So, little by little, better encryption support is being made available within the world of DB2 for z/OS.

From zJournal, Oct / Nov 2007
.

Home.