MAINFRAME CRYPTO

Database Encryption

Greg Boyd

gregboyd@mainframecrypto.com

www.mainframecrypto.com

November 2014

Copyrights and Trademarks

- Copyright © 2014 Greg Boyd, Mainframe Crypto, LLC. All rights reserved.
- All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. IBM, System z, zEnterprise and z/OS are trademarks of International Business Machines Corporation in the United States, other countries, or both. CA ACF2™ for z/OS and CA Top Secret® for z/OS are trademarks of CA, Inc. *Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries* All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.
- **THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY**. Greg Boyd and Mainframe Crypto, LLC assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will Greg Boyd or Mainframe Crypto, LLC be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if expressly advised in advance of the possibility of such damages.

Database Encryption

- How does it work - DB2 Built-In Functions
- How does it work – Guardium Infosphere Data Encryption Tool for IMS and DB2 (5799-P03)
- Comparisons
- Other Encryption

How do the DB2 Built-In Functions work?

- Under application control – you encrypt the fields that need to be secure
  - 'Password for Encryption' is hashed to generate a unique key
  - Hint can be used as a prompt for remembering the key
  - Encrypted field must be defined as VARCHAR (since it will contain binary data once its encrypted)
  - The encrypted field will be longer (next multiple of 8 bytes + 24 bytes of MetaData + 32 bytes for optional hint field)
  - TDES Only!

Encrypt (StringDataToEncrypt, PasswordOrPhrase, PasswordHint)
Decrypt_Char(EncryptedData, PasswordOrPhrase

DB2 Built-In Functions Example

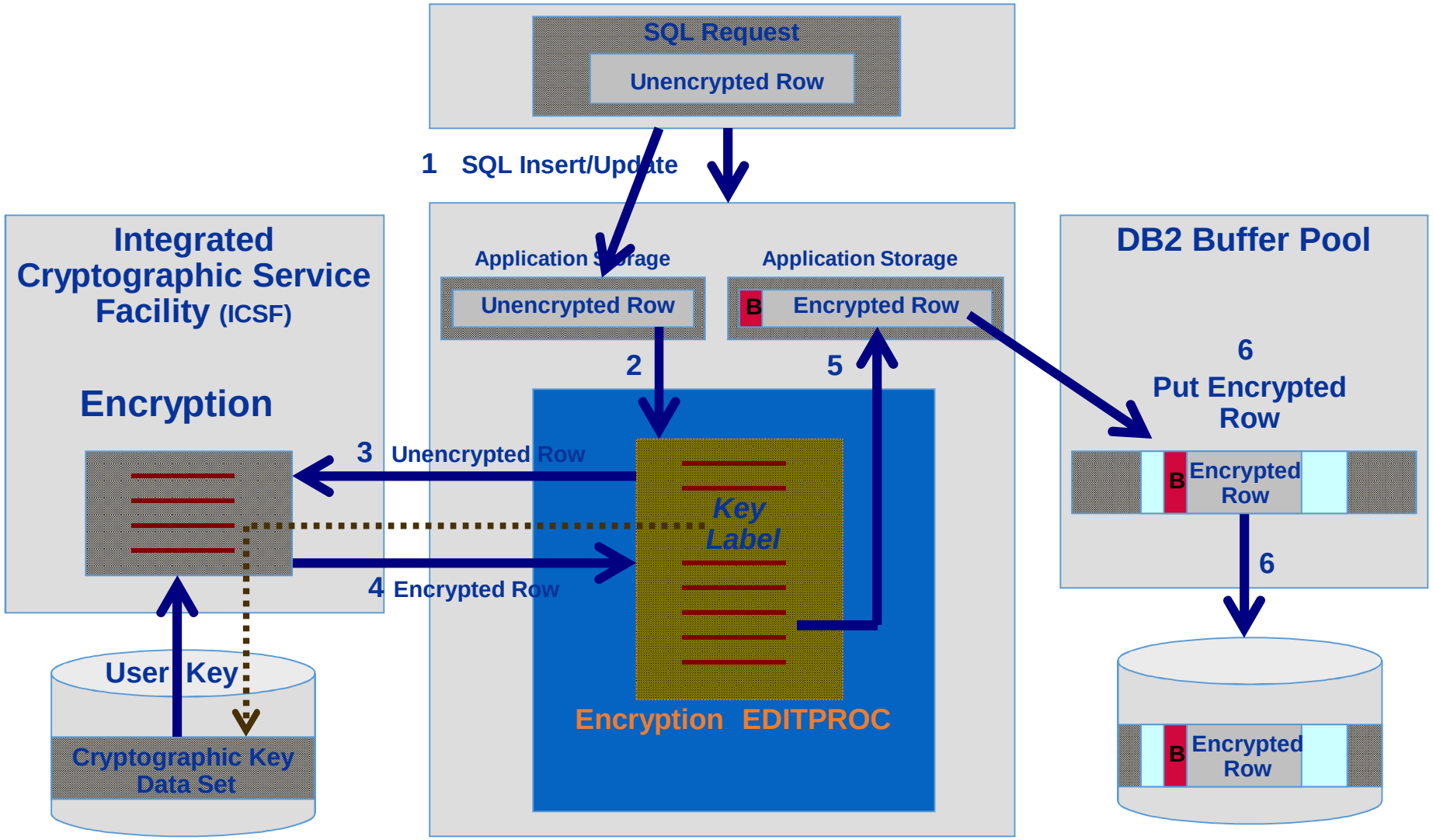
```
CREATE TABLE EMPL
(EMPNO VARCHAR(64) FOR BIT DATA,
EMPNAME CHAR(20),
CITY CHAR(20),
SALARY DECIMAL(9,2))
IN DSNDB04.RAMATEST ;

COMMIT;

SET ENCRYPTION PASSWORD = 'PEEKAY' WITH HINT 'ROTTIE';

INSERT INTO EMPL(EMPNO, EMPNAME, SALARY)
VALUES (ENCRYPT('123456'),'PAOLO BRUNI',20000.00) ;

INSERT INTO EMPL(EMPNO, EMPNAME, SALARY)
VALUES (ENCRYPT('123457'),'ERNIE MANCILL',20000.00) ;
```

From Redbook SG24-7959, Security Functions of IBM DB2 10 for z/OS

How does the Data Encryption Tool work?

- EDITPROC - for every row
  - Encrypted row same length as clear row
  - No application changes required
  - One key per table or segment specified in the EDITPROC
  - Indexes are not encrypted

DB2 Data Encryption Flow – Insert / Update



Enhancements to the Data Encryption Tool

- FIELDPROC – encrypts at the column level
  - Introduced by PM55879/UK76423
  - No application changes required
  - Indexes can be encrypted
  - One key, label specified in the FIELDPROC
  - Columns must be < 254 bytes; Column names must be < 18 chars in length
- UDF – User Defined Functions
  - Introduced by PM45364/UK72991
  - No application changes required; Minimally disruptive, columns encrypted in place
  - Indexes can be encrypted
  - One key, label specified in the UDF
  - All data types supported by UDFs can be encrypted
  - VIEW/TRIGGER – provides access control to the cleartext

Data Encryption Tool Algorithm restrictions

- DB2 EDITPROCs
  - DECENA00 – Clear Key AES or DES/TDES
  - DECENB00 – Protected Key AES or DES/TDES
  - DECENC00 – Secure Key AES or DES/TDES only
- DB2 FIELDPROCs
  - DECENF00 – Protected Key AES only
- DB2 UDFs
  - DECENU00 – Protected Key AES only
- IMS exit routines
  - DECENA01 – Clear Key DES/TDES
  - DECENB01 – Protected Key AES only
  - DECENC01 – Secure Key DES/TDES only

Cryptographic Keys

- Data Encryption Tool
  - Key must be stored in the CKDS
  - When the table with an EDITPROC/FIELDPROC is in use, the key is available in the DB2 address space
- DB2 BIF
  - Clear key only (it's calculated by hashing the password for encryption) – so it's available in the DB2 address space
  - Keys are not stored in a dataset, but the password for encryption is stored in the table

Changing Cryptographic Data Keys

- Data Encryption Tool
  - Unload, change EDITPROC/FIELDPROC to reference new key, reload
  - Unload, change current key, DB2 restart, reload
- DB2 BIF
  - Under application control

Database Indexes

- Index not encrypted
  - Encryption Tool EDITPROC – index is not encrypted (EDITPROC encrypts the entire row, so the data is encrypted, but the index is not)
    - Bad for security, good for performance

INDEX	SSN NAME ADDRESS
223491398	F{(œ(•´ú— GÿÞ# ¥†‰jÍiÑÆ

- Index encrypted
  - FIELDPROC - index can be encrypted
  - DB2 BIF - Application encrypts the field, if that field is an index, then the index is encrypted
    - Good for security, but may impact performance

INDEX	SSN NAME ADDRESS
F{(œ(•´ú	F{(œ(•´ú— GÿÞ# ¥†‰jÍiÑÆ

Data Encryption Tool – Hardware Requirements

- Clear Key
  - zEC12/zBC12 /z196/z114/z10 CPACF & crypto card for CKDS*
- Secure Key
  - zEC12/zBC12 /z196/z114/z10 Requires a crypto card
- Protected Key
  - z196/z114/z10 Requires a CEX3C
  - zEC12/zBC12 Requires a CEX4SC or CEX3C

*Prior to HCR7750 a crypto card is required to create and use a CKDS, beginning with HCR7751 ICSF supports a clear key only CKDS

**Protected Key support requires HCR7770 or higher

DB2 BIFs - Hardware Requirements

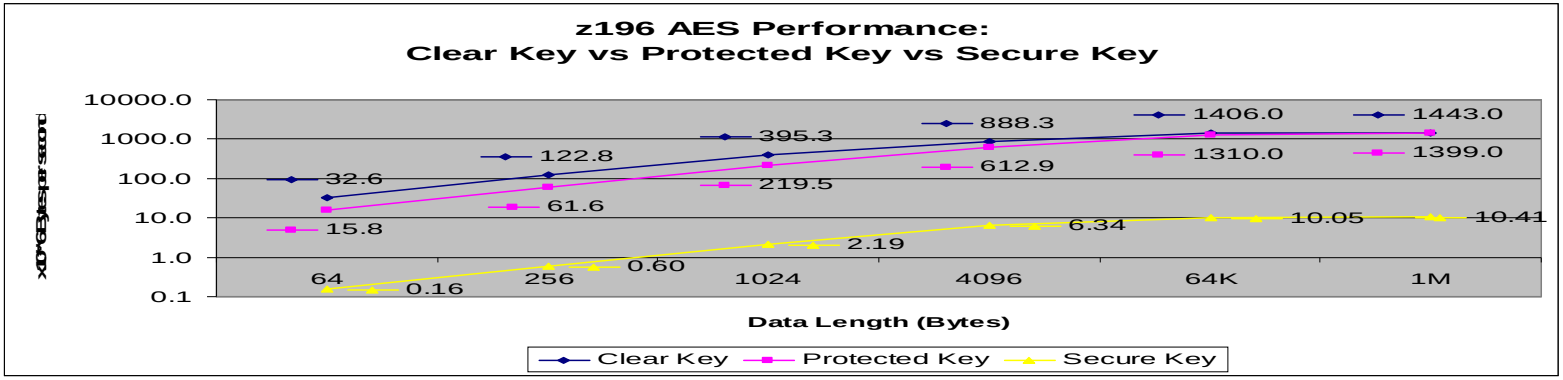
- zEC12/zBC12, z196/z114, z10 (CPACF)
  - Uses MSA instructions, not the ICSF APIs, but ICSF must be started to provide hashing support
  - TDES only

z196 Crypto Performance

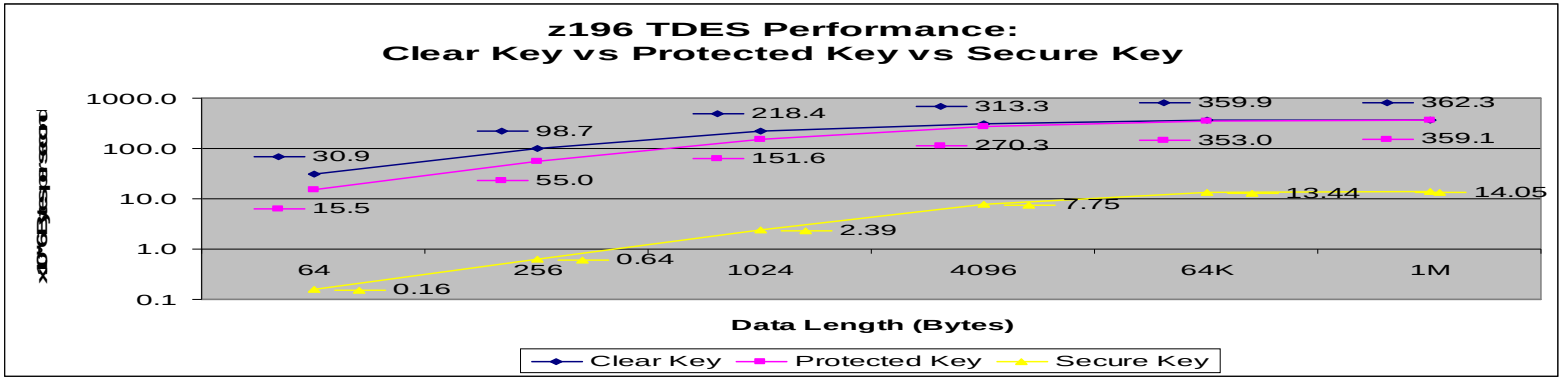
From the Crypto Performance Whitepapers

http://www.ibm.com/systems/z/advantages/security/z10cryptography.html

- AES Encryption



- TDES Encryption



Secure Key SQL Performance Results

```
JOBNAME:                        Current Thread Detail          DATE: 06/26/08
DB2 V8 :                                                       TIME: 11:42:17
COMMAND:                                                       CYCLE: MMSS

 CONN ID :                      PLAN    :          CURRENT STATE: INAPP
 CORR ID :                      AUTH ID :          THREAD START : 11:32:49.4763
 LOCATION:                      SQLID   :          CONN TYPE    : CALL ATTACH
 RQST LOC:                      LUWID   :          ?
 PKG LOC :                      ACCT TKN:
 PKG NAME:                   N.18386E190EC573D6
 +------------ Timings -------------+   +--------- Event Counts ----------+
 ELAPSED:  5:42.97  DB2 ELA:  0:02.96   WAIT    :      27   PACKAGES:     2
 TOT CPU:  0:00.14  DB2 CPU:  0:00.13   IFI     :       0   PARA GRP:     0
 I/O WT :  0:00.00- LOCK WT:  0:00.00   RMT CALL:       0   PARA CPU:     0
 SORT   :  0:00.00- TOT WT :  0:00.30   SORT    :       1   PARA MBR:     0
 NESTED :  0:00.00                      SQL LOGR:       0   DS OPENS:     1
                                        RID LIST:       0
 +---------- SQL Counts ----------+   +------ Buffer Pool/Locking ------+
 TOTAL  :     2054  PREPARES :      1   GETPAGE:       182  MX PG LK:     1
 SELECT :        1  OPEN CSR :      8   SYNC RD:         9  LOCKESCL:     0
 FETCH  :     2031  INCR BIND:      0   PREFTCH:         4  SUSPENDS:     0
 COMMITS:        2  SECURITY :      0   ASYN RD:         4  TIMEOUTS:     0
 DML    :        0  DDL      :      0   PGS/IO :      14.0  DEADLOCK:     0
```

Clear Key SQL Performance Results

```
DB2 V8 : DTVB                                                       TIME: 11:44:40
COMMAND:                                                            CYCLE: MMSS

 CONN ID :                  PLAN     :           CURRENT STATE: INAPP
 CORR ID :                  AUTH ID  :           THREAD START : 11:43:37.9172
 LOCATION:                  SQLID    :           CONN TYPE    : CALL ATTACH
 RQST LOC:                  LUWID    :
 PKG LOC :                  ACCT TKN:
 PKG NAME:                       :RTN.18386E190EC573D6
 +---------- Timings -------------+   +--------- Event Counts ----------+
 ELAPSED:  1:02.64  DB2 ELA:  0:00.36   WAIT     :      13   PACKAGES:     2
 TOT CPU:  0:00.03  DB2 CPU:  0:00.03   IFI      :       0   PARA GRP:     0
 I/O WT :  0:00.01  LOCK WT:  0:00.00   RMT CALL:        0   PARA CPU:     0
 SORT   :  0:00.00- TOT WT :  0:00.33   SORT     :       1   PARA MBR:     0
 NESTED :  0:00.00                      SQL LOGR:        0   DS OPENS:     1
                                        RID LIST:        0
 +----------- SQL Counts -----------+   +------ Buffer Pool/Locking ------+
 TOTAL   :      2054  PREPARES :     1   GETPAGE:       182  MX PG LK:     1
 SELECT  :         1  OPEN CSR :     8   SYNC RD:         3  LOCKESCL:     0
 FETCH   :      2031  INCR BIND:     0   PREFTCH:         4  SUSPENDS:     0
 COMMITS:          2  SECURITY :     0   ASYN RD:         4  TIMEOUTS:     0
 DML     :         0  DDL      :     0   PGS/IO :      26.0  DEADLOCK:     0
```

Secure vs. Clear Key: Database Load Results

Database utility loads of 200,000 rows yielded the following results:

(In seconds)	Clear Key	Secure Key
CPU Time	2	8
Elapsed Time	18	259

As you can see from the LOAD and SQL examples, secure key is considerably more CPU intensive.

Implementation–Example

Table xxx									
Encrypted Tables xxDBA					Non Encrypted Tables xxNON				
Utility	Elapsed Time	CPU Time	Init Date	Init Time	Utility	Elapsed Time	CPU Time	Init Date	Init Time
Unload	00:01:37.86	00:01:46.02	Sept. 28	9:15 A	Unload	00:01:42.64	00:01:08.31	Sept. 28	9:15 A
Load	00:04:07.73	00:03:45.13	Sept. 28	11:30 A	Load	00:03:40.55	00:03:12.89	Sept. 28	11:30 A
REORG	00:19:56.46	00:03:33.44	Sept. 28	2:30 P	REORG	00:05:49.17	00:02:12.37	Sept. 28	2:30 P
Index Rebuild	00:03:50.03	00:01:32.04	Sept. 29	9:00 A	Index Rebuild	00:01:20.30	00:00:48.94	Sept. 29	9:00 A
Image Copy	00:07:05.19	00:00:08.10	Sept. 29	1:00 P	Image Copy	00:03:51.43	00:00:07.56	Sept. 29	1:00 P
Recover	00:07:05.19	00:00:08.10	Sept 29	2:15 P	Recover	00:03:51.43	00:00:07.56	Sept. 29	2:15 P
DSNTIAUL	00:05:42.22	00:04:31.99	Sept. 30	9:30 A	DSNTIAUL	00:05:23.32	00:03:52.42	Sept. 30	9:30 A

Your mileage may vary.

Protected Key – Internal Benchmark

- Performed on a z10 running under ICSF release HCR7770. The benchmark was a Load Replace of 16.4 million rows, and was using AES encryption, all numbers are minutes:seconds of CPU. Tests were run using CHECKAUTH settings of Yes and No:
  - Clear Key - KMC instruction - 1:15
  - Protected Key - CHECKAUTH (NO) - 4:00
  - Protected Key - CHECKAUTH(YES) -4:38
  - Secure Key - CHECKAUTH(NO) - 17:34

Side-by-side Comparison

	Column (DB2 Built-In Functions)	Row/Table (IBM Encryption Tool for IMS and DB2)
DB2 Support	▪V8, V9, V10 ▪Data in indexes is encrypted ▪Does not work w/DB2 Load Utility ▪Data type of encrypted columns must be FOR BIT DATA	▪V7.x, V8.x, V9.x, v10.x ▪DB2 index data is not encrypted. ▪Works with all DB2 utilities
Application Change Required	▪Application must change to invoke the BIFs for the columns that will be encrypted	▪No application change, but each table will need to be recreated with an EDITPROC
Transaction Processing Overhead	▪The cost overhead depends on hardware, DB2 and application access	▪High overhead due to the amount of data encryptions
Key Management	▪Application has responsibility for the encryption key	▪Keys are managed by and accessed through ICSF
Pre-Reqs	▪ICSF must be active ▪CPACF hardware	▪ICSF must be active ▪Secure PCI card, unless running HCR7751 or later and clear key only CKDS

Decisions, Decisions …

- Ownership (i.e. politics)
  - Data Administrator - Data Encryption Tool
    - Sets up the EDITPROC and specifies the key to be used for the entire table
    - Key must be defined to/managed by ICSF (stored in the CKDS)
  - Application - DB2
    - Application logic determines which key to use for each field/column
    - Password is managed by the application
- Security requirements
- Performance requirements
- Application/production support
- Space considerations
- Crypto hardware available

Other DB2 Encryption

- Between DB2 databases
  - zIIP Assisted IPSec (VPN) on z/OS
- DASD Encryption
  - Protects the data when the DASD leaves your control, it does not protect the data from internal users
- Tape Encryption
  - Log files
  - Database unloads

Closing Thoughts

- Encryption has a cost
  - Crypto hardware more efficient with large blocks of data
- Secure Key on a PCI Card – more expensive
- Clear Key exists in the DB2 Address Space, Protected Key and Secure Key are too, but they are stored encrypted under the Wrapping Key or Master Key

Data Encryption for Databases - Reference Materials

- SC19-3219 IBM Infosphere Guardium Data Encryption for DB2 and IMS Databases Version 1 Release 2 User Guide
- Redbooks
  - SG24-6465 DB2 UDB for z/OS Version 8 Performance Topics
  - SG24-7959 Security Functions of IBM DB2 10 for z/OS (Sept. 2011, doesn't cover FIELDPROCs and UDFs)
- Articles
  - IMS Newletter article: "Encrypt your IMS and DB2 data on z/OS"- ftp://ftp.software.ibm.com/software/data/ims/shelf/quarterly/fall2005.pdf
  - IBM Infosphere Guardium Data Encryption for DB2 and IMS Databases – new solutions for DB2 - http://www-304.ibm.com/support/docview.wss?uid=swg21586761&aid=1

Questions?