Database Encryption

Greg Boyd

gregboyd@mainframecrypto.com

www.mainframecrypto.com

Unscrambling the Complexity of Crypto!

Nov 2016

Copyrights and Trademarks

- **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
- Performance
- 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 (using MD5) 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

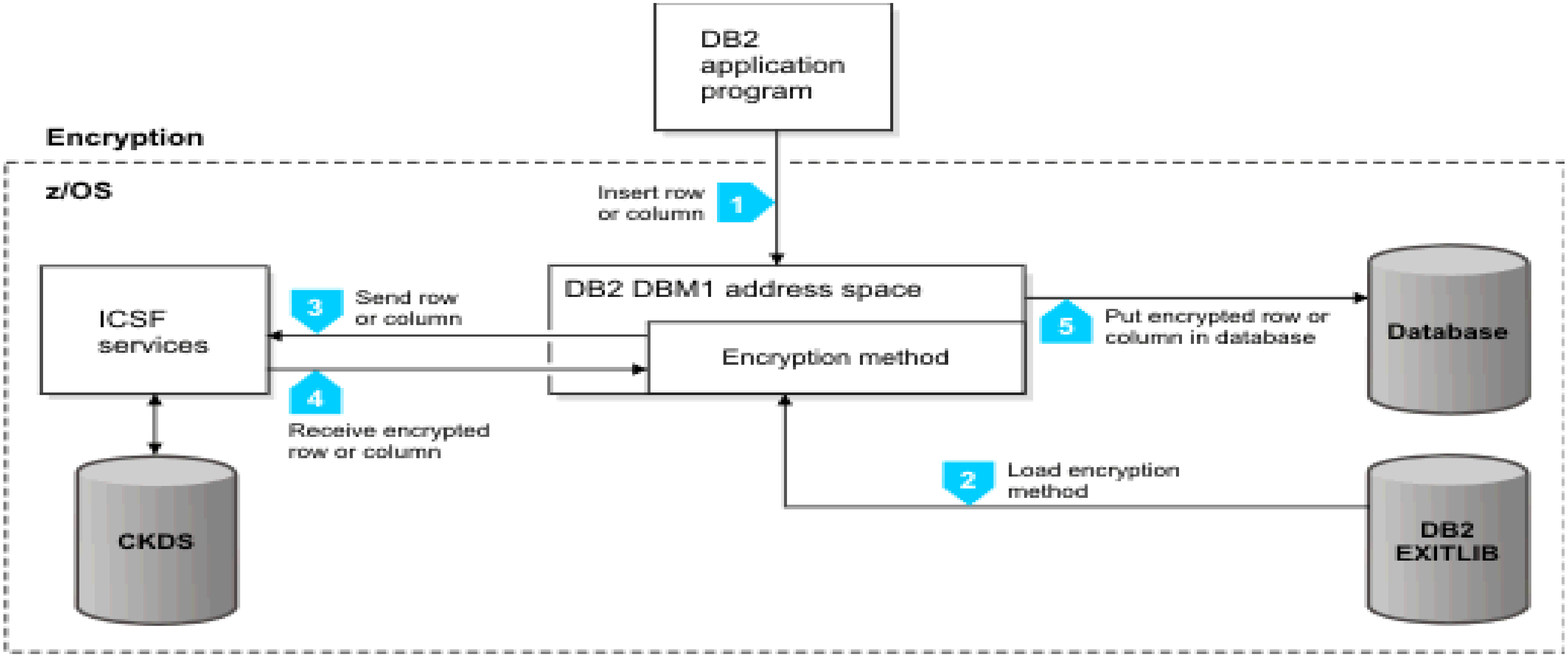
How does the Data Encryption Tool work?

- 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

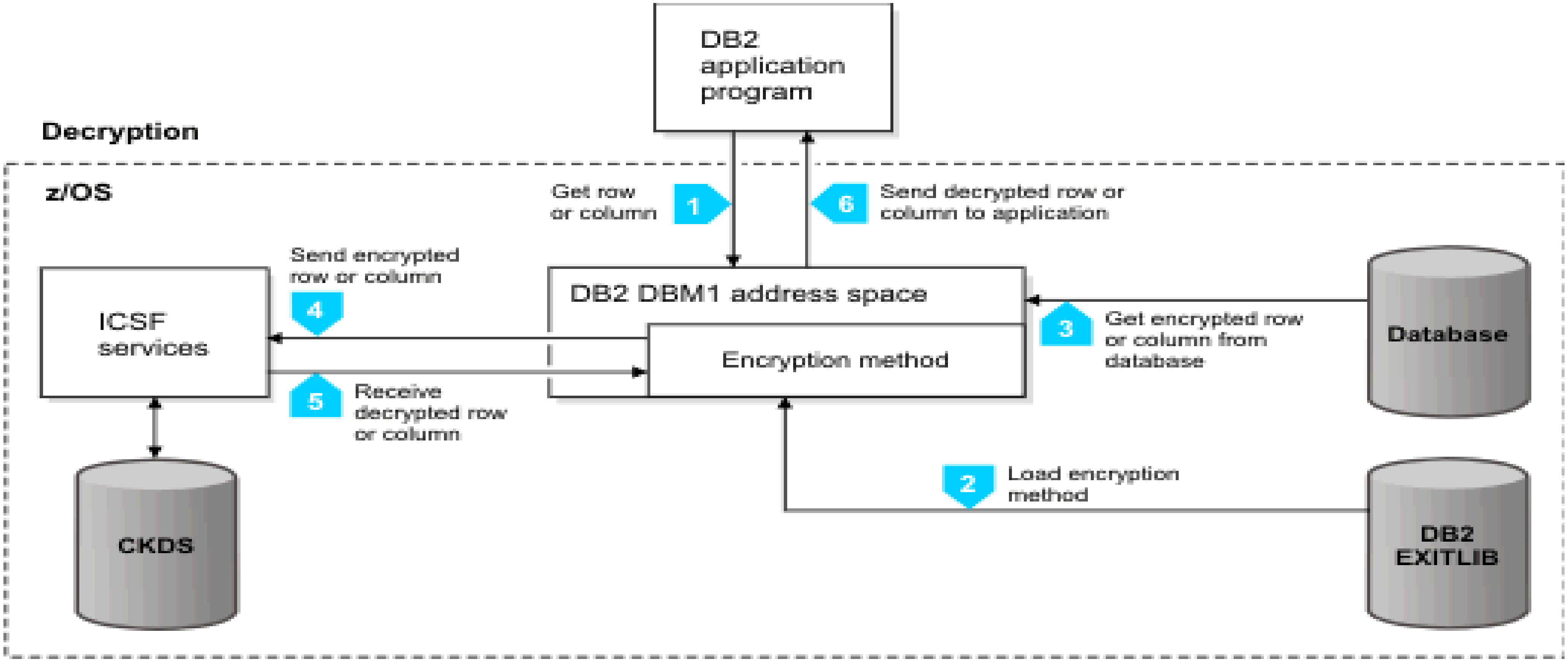
DB2 column encryption

- FIELDPROC – encrypts at the column level
  - 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
  - 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

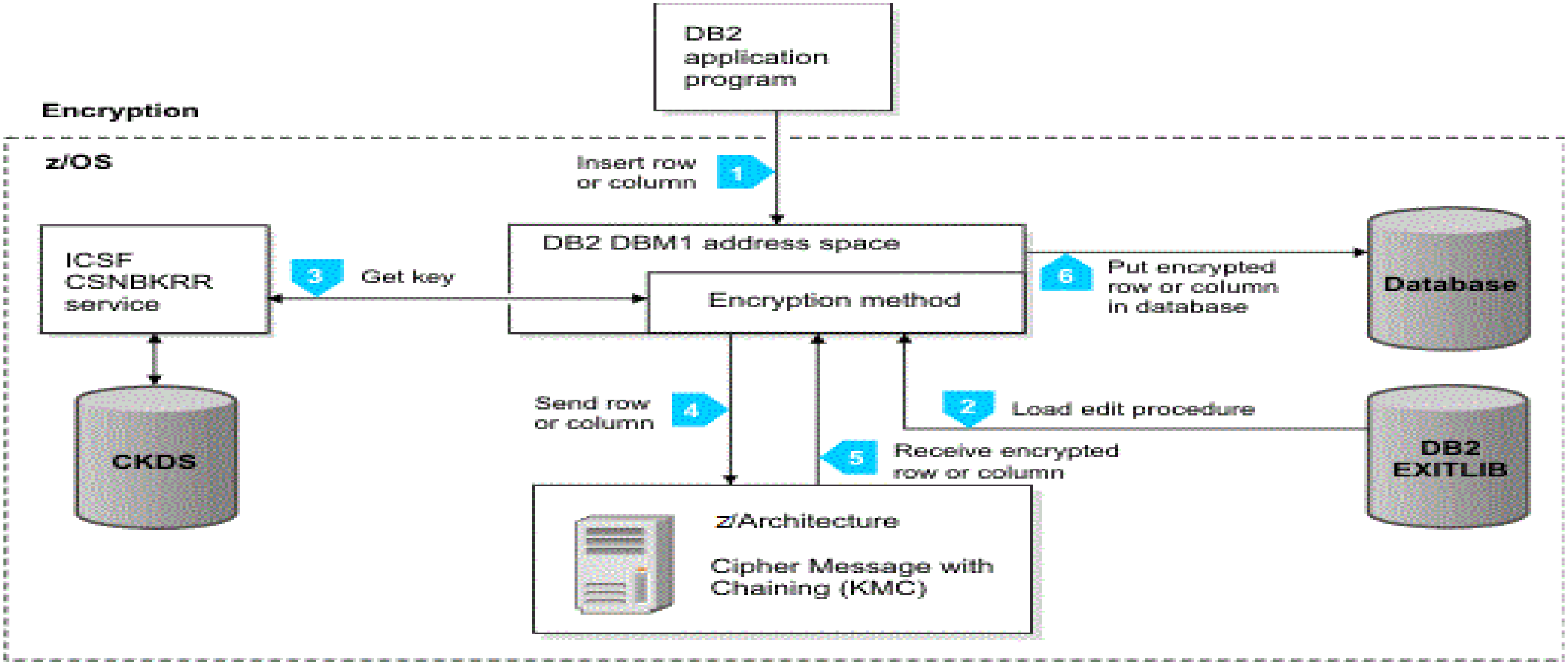
DB2 encryption flow



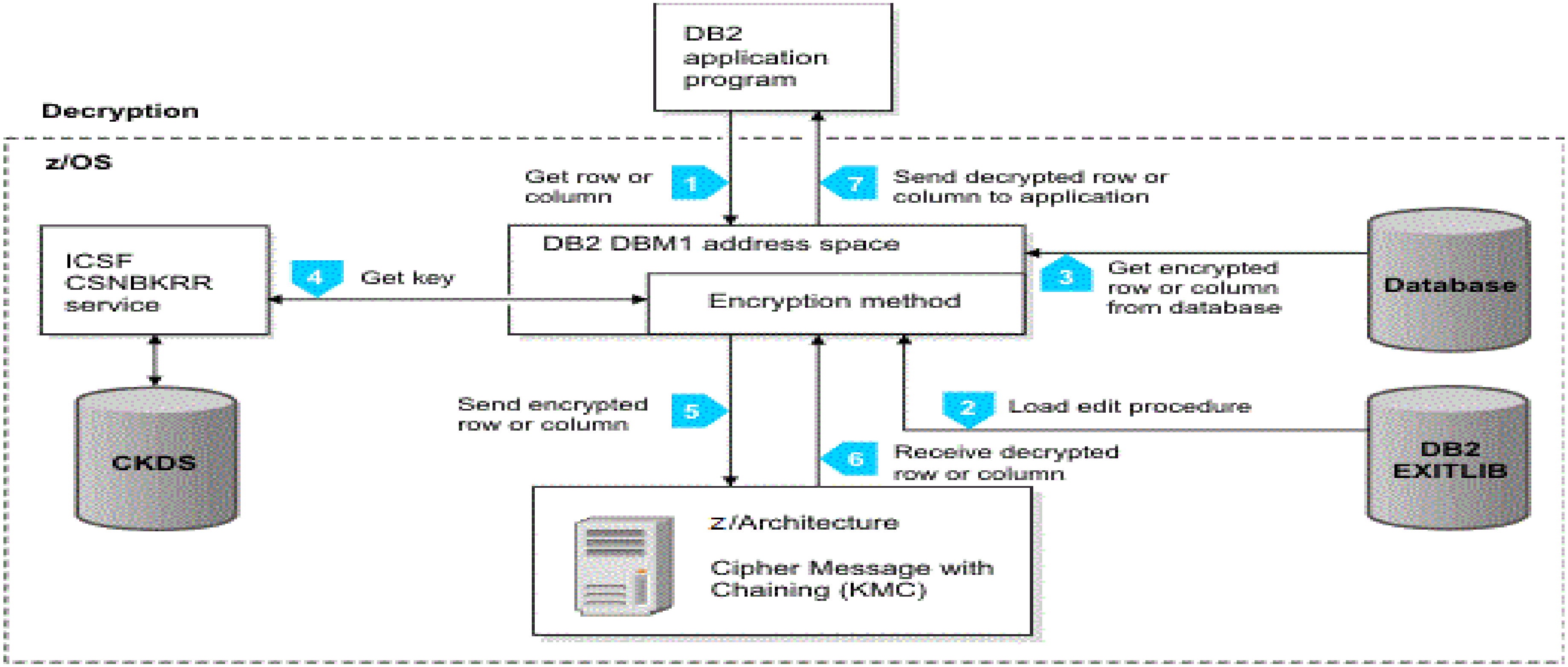
DB2 decryption flow



DB2 encryption flow using clear keys



DB2 decryption flow using clear keys



Protects data within the database infrastructure

- DB2 and IMS databases
- Image copy datasets
- DASD volume backups

Implementing an EDITPROC

- Generate Key using ICSF KGUP (Key Generation Update Program)
- Prepare EDITPROC using Data Encryption Tool providing ICSF Keylabel
- Unload target table
- DROP / RECREATE table specifying EDITPROC
- LOAD table
- Done!

DB2 Exit Routines

	EDITPROC	Description	Implementation	Algorithms
DB2 EDITPROC	DECENA00	Clear Key	CSNBKRR & native instructions	AES, TDES/DES
	DECENAA0 (PI58257)	Clear Key/KMO	CSNBKRR, KMO + native instructions	AES, TDES/DES
	DECENB00	CPACF Protected Key	CSNBSYE/CSNBSYD	AES
	DECENBI0	CPACF Protected Key plus Unique ICV Generation	CSNBSYE/CSNBSYD	AES
	DECENC00	Secure Key	CSNBENC/CSNBDEC	TDES/DES
	DECENCA0	Secure Key plus AES	CSNBSAE/CSNBSAD	AES
DB2 FIELDPROC	DECENF00	CPACF Protected Key	CSNBSYE/CSNBSYD	AES
DB2 UDF	DECENU00	CPACF Protected Key w/default IV	CSNBSYE/CSNBSYD	AES
	DECENUI0	CPACF Protected Key w/unique ICV	CSNBSYE/CSNBSYD	AES
	DECENUP0	CPACF Protected Key w/unique ICV & padding	CSNBSYE/CSNBSYD	AES
	DECENURN	Generate unique ICV	CSNBRNGL	
	DECENUBL	CPACF Protected Key w/unique ICV for BLOBs & Large objects	CSNBSYE/CSNBSYD	AES

IMS Exit Routines

IMS Exit Routines	DECENA01	Clear Key	CSNBENC/CSNBDEC	TDES/DES
	DECENAA1 (PI57908/ UI37167)	Clear Key with CPACF protected key wrapping Batch ICSF CHECKAUTH recurring bypass	CSNBKRR	AES,TDES/DES
	DECENB01	CPACF protected key	CSNBSYE/CSNBSYD	AES
	DECENBB1 (PI55772/ UI38988)	CPACF protected key with batch ICSF CHECKAUTH recurring bypass	CSNBSYE/CSNBSYD	AES
	DECENC01	Secure key	CSNBENC/CSNBDEC	TDES/DES

Compression before encryption

- Compression & encryption driver routine
  - DECENADV - Driver
  - DECZLDX0 – Compression

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

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

Data Encryption Tool – Hardware Requirements

- Clear Key
  - z13/z13s, zEC12/zBC12, z196/z114,z10, z9
    - CPACF only, no crypto card is required IF using a clear key only CKDS
- Secure Key & Protected Key**
  - z13/z13s, zEC12/zBC12, z196/z114, z10
    - Requires a crypto card

*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

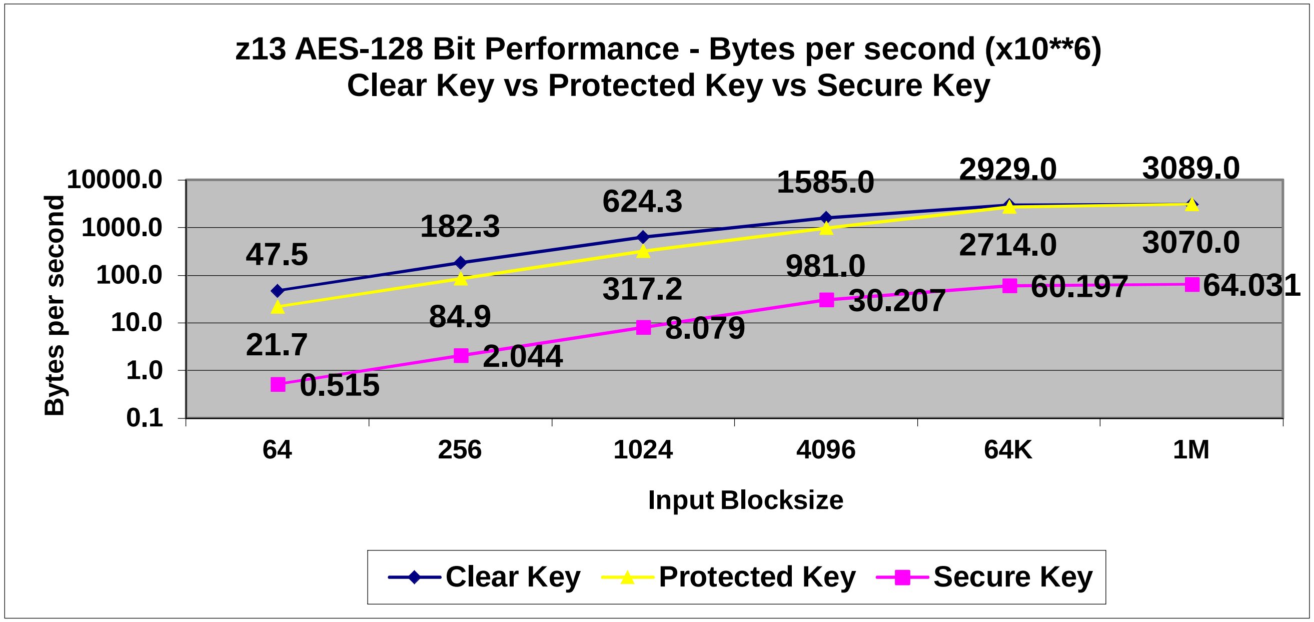
DB2 BIFs - Hardware Requirements

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

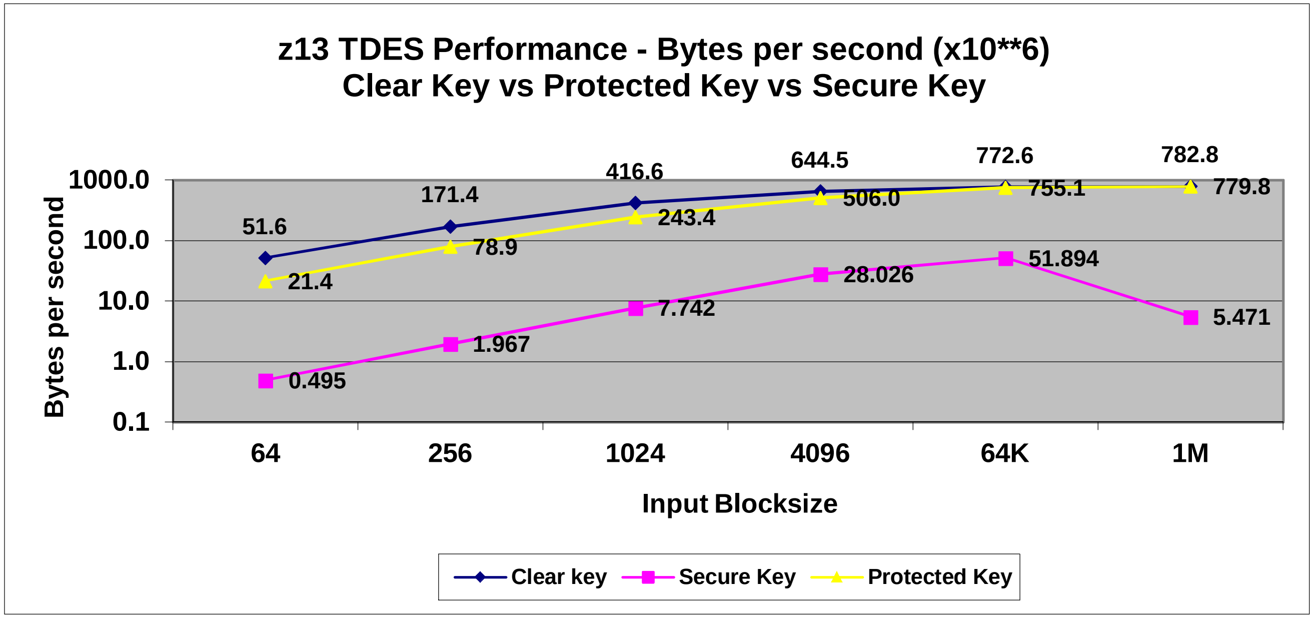
z13 Symmetric Key Performance

- Adapted from the IBM z13 Cryptographic Performance March 2015 document at

http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&htmlfid=ZSW03283USEN&attachment=ZSW03283USEN.PDF



z13 AES-128 Bit Performance - Bytes per second (x10**6)
Clear Key vs Protected Key vs Secure Key



z13 TDES Performance - Bytes per second (x10**6)
Clear Key vs Protected Key vs Secure Key

A note on statistics

Mark Twain said

“There are three kinds of lies: lies, damned lies and statistics.“

- Tom’s translation:

“Pick a number between 1 and 2000 and I will find an SQL statement where the act of encrypting the data will cause CPU usage to increase by that percentage.”

Slide courtesy of Tom Hubbard, Rocket Software

Hardware and Software

- Hardware:
  - z13 - 2964-609
  - CryptoExpress CEX5S card
- Software
  - z/OS version - 2.2
  - ICSF version - HCR77B1
  - DB2 version 11
  - Encryption Tool for DB2 and IMS Databases version 1.2
  - All systems with current maintenance
- TDES uses 192 bit key
- AES used 256 bit key

Slide courtesy of Tom Hubbard, Rocket Software

SQL Workload

- 14 different queries
- Accessing two main tables
  - DECPERFM.CQM32_SUMM_OBJECTS
  - DECPERFM.CQM32_SUMM_METRICS
- A third table is used for some joins
  - DECPERFM.CQM32_STMT_TYPES
    - Used to convert a SMALLINT value to a text literal for reporting

Slide courtesy of Tom Hubbard, Rocket Software

Relevant Table Statistics (1)

DECPERFM.CQM32_SUMM_METRICS

- Maximum row length : 1409
- Number of columns : 174
- Row count . . . . . : 53875
- Occupied pages . . : 17959
- Pct TS pages w/rows: 99
- Average row length : 1031

DECPERFM.CQM32_SUMM_OBJECTS

- Maximum row length : 766
- Number of columns : 48
- Row count . . . . . : 230206
- Occupied pages . . : 16458
- Pct TS pages w/rows: 99
- Average row length : 268

Slide courtesy of Tom Hubbard, Rocket Software

Relevant Table Statistics (2)

DECPERFM.CQM32_STMT_TYPES

- Object ID for table: 60
- Maximum row length : 44
- Row count . . . . : 300
- Occupied pages . . : 3
- Pct TS pages w/rows: 75
- Average row length : 32
- Stats feedback . . : Yes

Slide courtesy of Tom Hubbard, Rocket Software

Query Text

QUERY014:

```
    SELECT SMFID
      ,CQM_SUBSYSTEM
      ,INTERVAL_NUMBER
      ,INTERVAL_START
      ,METRICS_TOKEN
      ,METRICS_TIMESTAMP
      ,DBID
      ,OBID
      ,PSID
      ,BUFFERPOOL_NORM
      ,BUFFERPOOL_NUM
      ,OBJECT_TYPE
      ,DATABASE_NAME
      ,PAGESET_NAME
     FROM DECPERFM.CQM32_SUMM_OBJECTS
     ORDER BY DBID
        ,OBID
        ,PSID;
```

QUERY012:

```
              SELECT * FROM
DECPERFM.CQM32_SUMM_OBJECTS ;
```

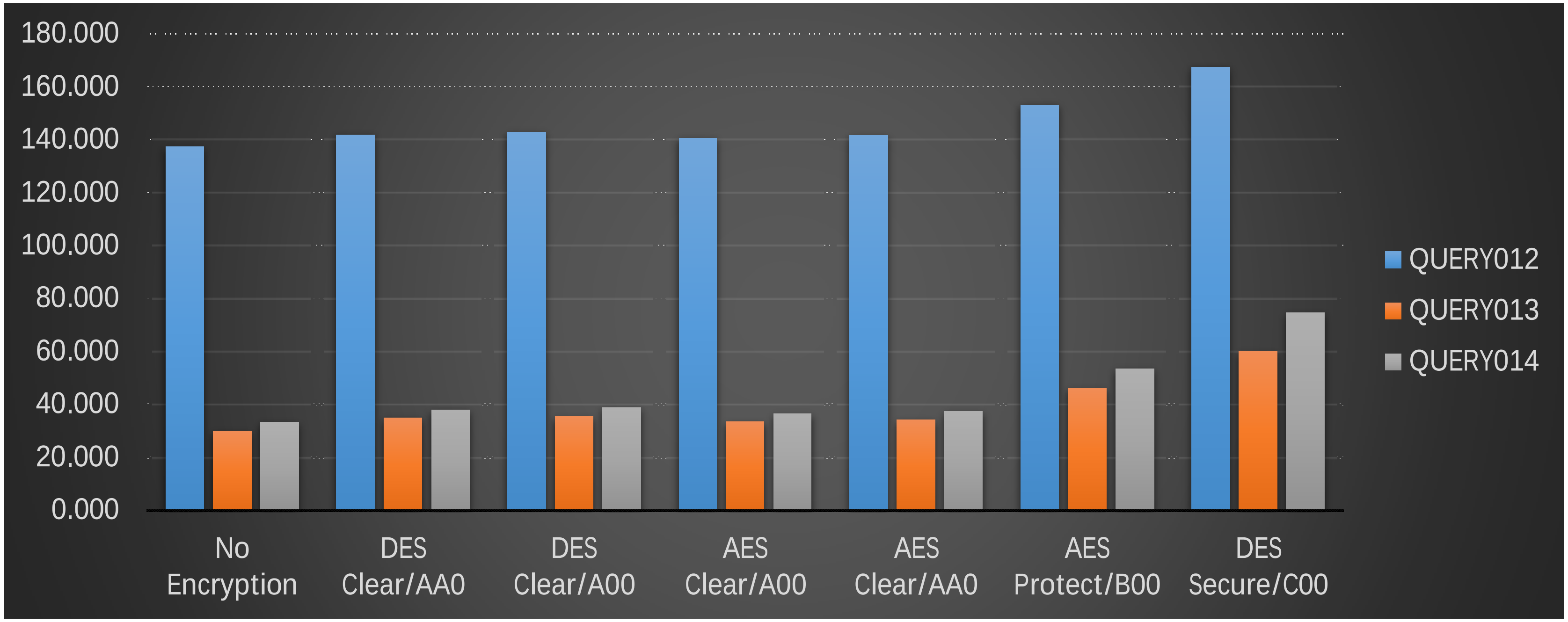
QUERY013:

```
    SELECT SMFID
      ,CQM_SUBSYSTEM
      ,INTERVAL_NUMBER
      ,INTERVAL_START
      ,METRICS_TOKEN
      ,METRICS_TIMESTAMP
      ,DBID
      ,OBID
      ,PSID
      ,BUFFERPOOL_NORM
      ,BUFFERPOOL_NUM
      ,OBJECT_TYPE
      ,DATABASE_NAME
      ,PAGESET_NAME
     FROM DECPERFM.CQM32_SUMM_OBJECTS ;
```

Slide courtesy of Tom Hubbard, Rocket Software

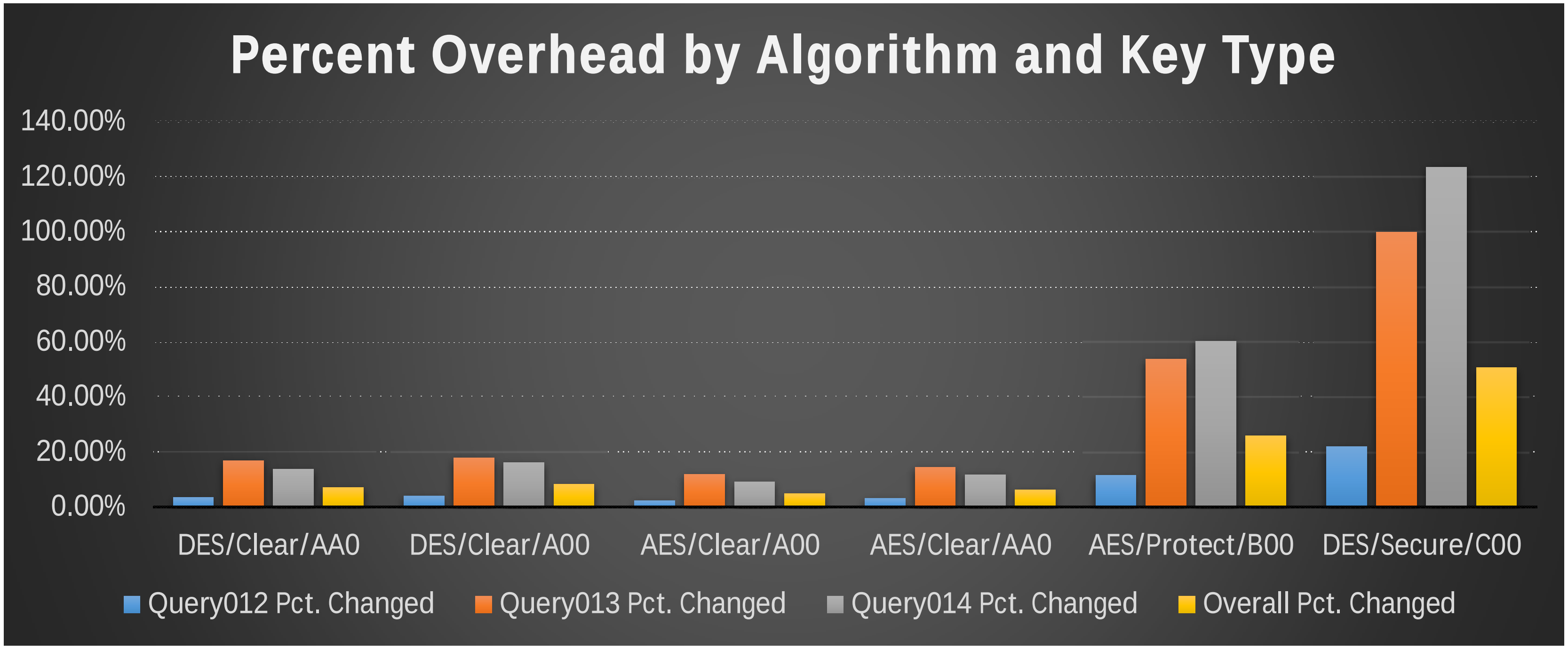
Relative Cost for Selected Queries

Note: Each query executed 30 times.



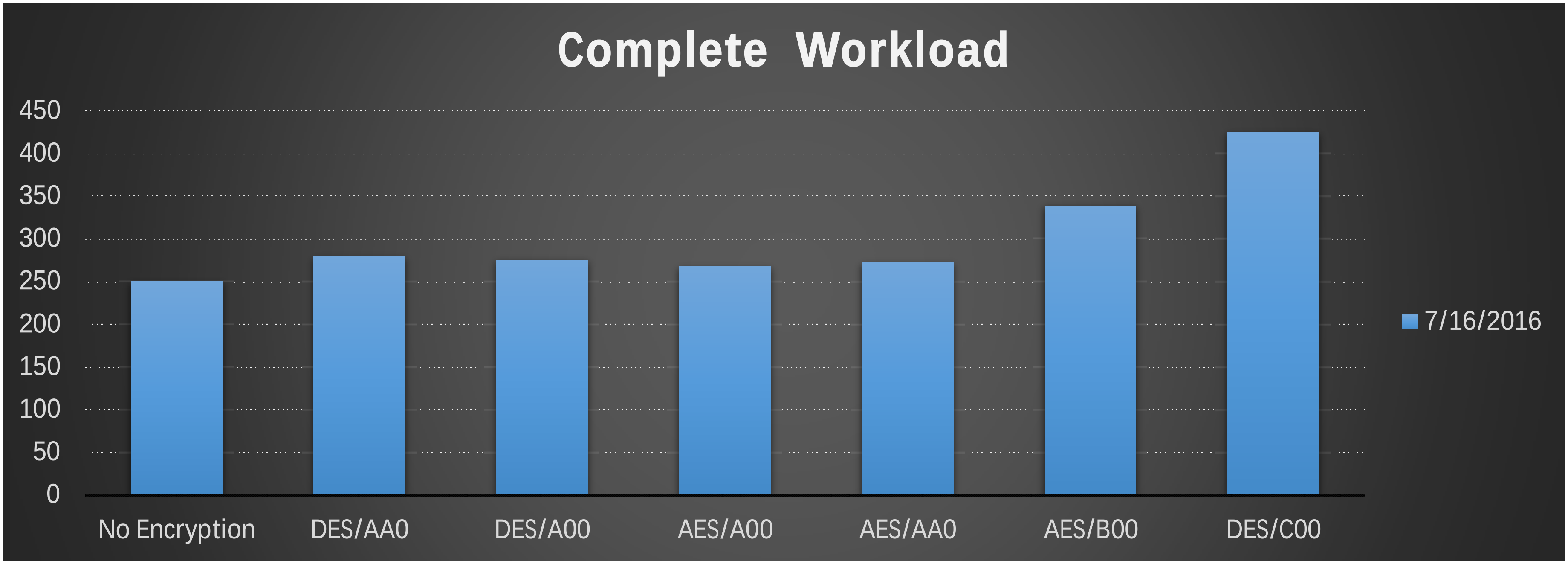
Slide courtesy of Tom Hubbard, Rocket Software

Percent of Overhead Increase



Slide courtesy of Tom Hubbard, Rocket Software

CPU Consumption for the Complete SQL Workload



Slide courtesy of Tom Hubbard, Rocket Software

When we look at percentages?

	Query012 Pct. Changed	Query013 Pct. Changed	Query014 Pct. Changed	Overall Pct. Changed for 3 Queries	Overall Pct. Changed for 14 Queries
DES/Clear/A00	3.39%	16.83%	13.65%	7.11%	10.19%
DES/KMO/AA0	4.04%	17.93%	16.18%	8.14%	11.59%
AES/Clear/A00	2.42%	11.95%	9.08%	4.95%	6.98%
AES/KMO/AA0	3.19%	14.39%	11.85%	6.31%	8.88%
AES/Protect/B00	11.62%	53.67%	60.29%	26.01%	35.49%
DES/Secure/C00	22.04%	99.95%	123.56%	50.59%	69.89%

Slide courtesy of Tom Hubbard, Rocket Software

Performance Conclusions

- Encryption adds to CPU usage
  - You may need to retune applications based on the performance impact of encryption
- Encryption overhead is reported in the DB2 accounting class 1 CPU and elapsed times
- AES encryptions adds slightly less CPU than TDES
- Protected keys add more overhead than clear keys
- Secure keys add a lot more overhead than protected keys
  - Secure keys also dramatically increase elapsed times
- The more CPU consumed by business logic (class 1) and DB2 (class 2) processing per row, the lower the % increase in relative overhead to encrypt the data.

Slide courtesy of Tom Hubbard, Rocket Software

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

Data Encryption for Databases - Reference Materials

- SC19-3219 IBM Infosphere Guardium Data Encryption for DB2 and IMS Databases Version 1 Release 2 User Guide
- Tom Hubbard Share Presentation – Database Encryption on z/OS
  - http://www.share.org/p/do/sd/topic=566&sid=12685
- Articles
  - Best Practices for Implementing IBM Data Encryption for DB2 and IMS Databases
    - http://publibfp.dhe.ibm.com/epubs/pdf/c2790010.pdf
  - Database encryption using IBM InfoSphere Guardium for DB2 and IMS
    - https://developer.ibm.com/zsystems/2016/06/17/database-encryption-using-ibm-infosphere-guardium-for-db2-and-ims/
  - IMS Newletter article: "Encrypt your IMS and DB2 data on z/OS"
    - ftp://ftp.software.ibm.com/software/data/ims/shelf/quarterly/fall2005.pdf

Data Encryption for Databases - Reference Materials

- 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)
  - SG24-7720 Securing and Auditing Data on DB2 for z-OS

Questions?