IBMi Encryption

Paul ROY, Real Solutions

Agenda

- Basic of Cryptography
- IBM i Capabilities
  - ASP Encryption
  - SSL
  - Key Management
- IBM i Software
  - Using FIELDPROC
  - BRMS
  - SQL

Basics of Cryptography

- Encryption

converting data in some unreadable form to protect privacy.

Decryption is the reverse process.

Encryption and decryption requires some extra data called a key.

Basics of Cryptography

- Authentication

ensures that the message was originated from the originator claimed in the message.

The message is « signed ».

- Integrity

ensures that the message sent is not altered.

Basics of Cryptography

- Non repudiation

ensures that the originator cannot deny the message.

The message is « digitally signed ».

Types of Cryptography

- Secret key Cryptography

Symmetric encryption: The same key is used to encrypt and decrypt.

- Public key cryptography

Assymmetric encryption: requires a pair of key.

- Hash Functions

algorithms that compute a value based on the message content.

Sample symmetric Key

- Algorithm
  - Add Key value to each Character in message to encrypt
  - Sub Key value from each Character in message to decrypt
- Example
  - Message : « secret data »
    - X'A285839985408481A381'
    - ADD KEY : x'02 »
    - X'A487859B87428683A583'
    - « uge gv fcvc »

Assymetric key

- To simple fake example
  - MyPrivateKey + YourPublicKey = EnccryptionKey
  - Key Algorithm = XOR
  - Prereq for key:
    MyPrivKey+YourPubKey = MyPubKey+YourPrivKey
  - Keys= (3 , 6) (4,5)
  - Message : « secret data »
    - X'A285839985408481A381'
    - XOR 09 ➔'AB8E8C908E498D8AAC8A'

Assymetric Key sample (RSA)

- RSA (River, Shamir, Adleman) algorithm
- **K = M^E mod N**
- **M = K^D mod N**
  - **K = Krypted message**
  - **M = Message**
  - **(E,N) = Public Key**
  - **(E,D) = Private Key**

RSA

- Create the key pair
  - Chose 2 prime number P, Q (ex: P=61, Q=53)
  - Compute N = P*Q = 61*53= 3233
  - Chose E with no common factor with (P-1)*(Q-1)
    - (61-1)*(53-1) = 3120
    - → E= 17
  - Compute D : E*D mod 3223= 1 ➔D=2753
    - PUBLIC KEY ➔ (17,3223)
    - PRIVATE KEY ➔ (2753,3223)

RSA mechanism

- Encryption :
  - Example to encrypt the value m=65
  - c = 65^17 mod 3233 = 2790
- Decryption
  - To decrypt 2790
  - M=2790^2753 mod 3233 = 65

Signature

- Signature is provided by a hash algorithm
  - Easy to compute
  - Impossible to generate message from hash
  - Infeasible to modify message without changing the hash
  - Infeasible to find two different messages with the same hash

HASH

- Used to store passwords
- HASH + SALT
- Avoid to hash the same value

- SHA-1, MD-5, SHA-2, MDC,etc…

PKI

- Public key cryptography = Assymetric cryptography.
- Certificates
  - Contain
    - Identity
    - Public and private key
  - are trusted /signed
    - By a Certificate Authority (isssues certificates)
  - Used for SSL (encrypt TCP/IP data transmission)

KEYSTORE

- A key store is a place to store keys and usually certificates…
- Can be a PF in a library or a stream file, a data area, a user space, etc…
- A Truststore is a key store to store trusted certificates (certificates of C.A that allows you to trust other)

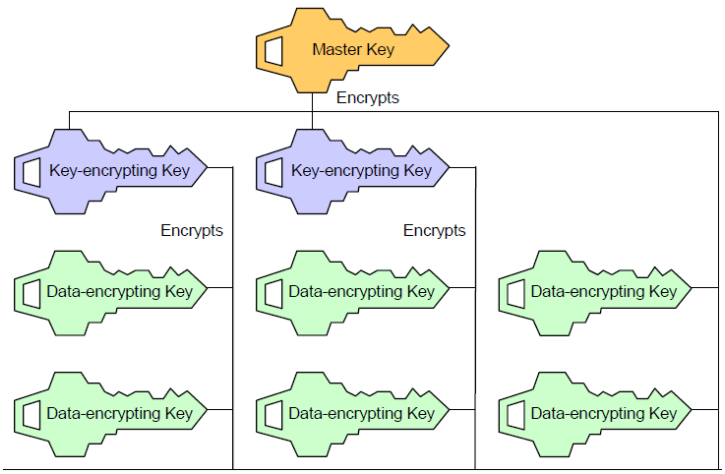
SSL

- SSL is based on KEY stored in certificates.
- Certificates must be trusted.
- The communication starts with an exchange of certificates
- 1) If the certificates are trusted, (issued or signed by a CA that is trusted (the CA cert is installed)
- 2) Both parties agree (discuss on data encrypted assymetricaly) on a symmetric key that will be used to encrypt data.

Key management

- If key is lost data is lost…
- Key must be encrypted…
  -> key hierarchy
- If key is weak… data is at risk
- Size matters !!
  – strong protection requires long key
  – Long key impact performance

Key hierarchy



Key management

- Keys must be saved each time they are changed.
- Keys must be protected (restricted to authorized users)

MASTER KEY

- IBM i uses 8 master keys.
- Each master key has 3 versions
  - New
  - Current
  - Old
- Master keys are used to encrypt other keys (KEK and data keys) not data.

Key management

- IBM i provides GUI interface ,CL commands and API's to manage the master keys
  - Load Master key :enter passphrases that are xored and hashed to create a new version
  - Set Master Key
    - Move the current to the old version
    - Move the new version to the current
    - Clear the new
    - Return KVV (Key verif value)
  - Test Master Key
    - Return KVV

KEY STORE and Keys

- IBM i provides web interface and API to
  - Create key store
  - Add key to key store
- When master Key is changed all keys and keystore must be translated

DATA BASE encryption

- Field consideration
- A cipher algorithm has minimum lenght requirement
- Must be character field
- CCSID must be 65535
- Exemple a national number (11 digits) is usually stored in 6 packed bytes but encryption requires 16 characters.

Data base consideration

- If the field is changed
  - Need migration/conversion program
  - Recreate all pgm
- The encrypted field could be stored in another file.
- SORT on encrypted data produces unpredictable result…
- SEARCH might be very slow…. (need full decryption…)

FIELDPROC

- New in IBM i 7.1
- Program (exit point) called by data management function every time a record is read, written or updated..
- Added by SQL
  ALTER TABLE CreditCards ALTER COLUMN CredCardNum SET FIELDPROC ToolsLib/FldPgm
- *The feature is standard in IBM i and can be used for encryption but you have to write (or buy) the encryption software.*

FIELDPROC

- Performance
  - Multiple fieldproc in one record means multiple pgm calls…

EXAMPLE

- CREATE TABLE COMMON/CREDCARD1 (ACCOUNTNBR DEC (7) NOT NULL WITH DEFAULT, CLIENTNAME CHAR (64) NOT NULL WITH DEFAULT, CARDNBR CHAR (12) NOT NULL WITH DEFAULT)
- ALTER TABLE common/CREDCARD1 ALTER COLUMN CARDNBR SET FIELDPROC COMMON/FLDPRC003

Example

- FIELDPROC program

SQL

```
CREATE TABLE COMMON/CREDCARD3
(ACCOUNTNBR DEC ( 7) NOT NULL WITH
DEFAULT,
CLIENTNAME CHAR (128) for bit data
 NOT NULL WITH DEFAULT,
CARDNBR CHAR
( 12) NOT NULL WITH DEFAULT FIELDPROC
COMMON/FLDPRC003 )
```

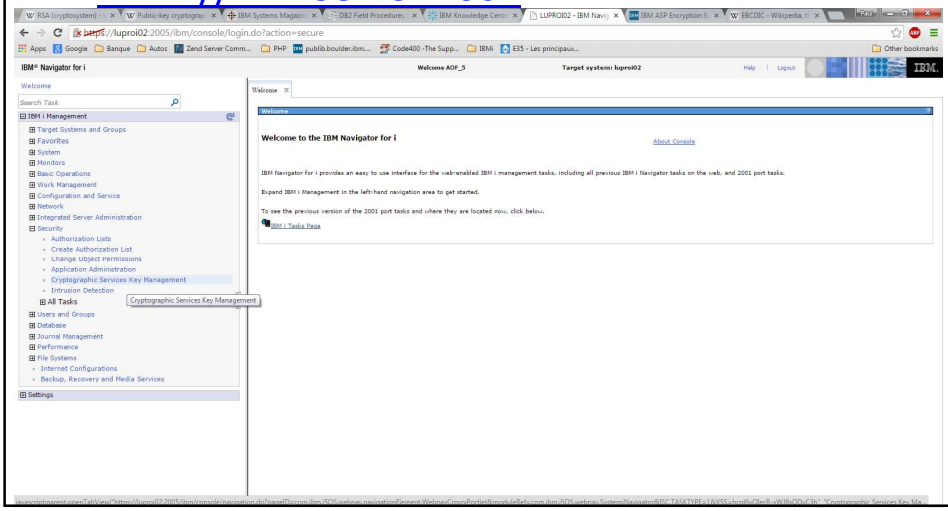
SQL

- INSERT INTO COMMON/CREDCARD3 VALUES(101, ENCRYPT_AES('PAUL ROY') ,
- '220033004400')

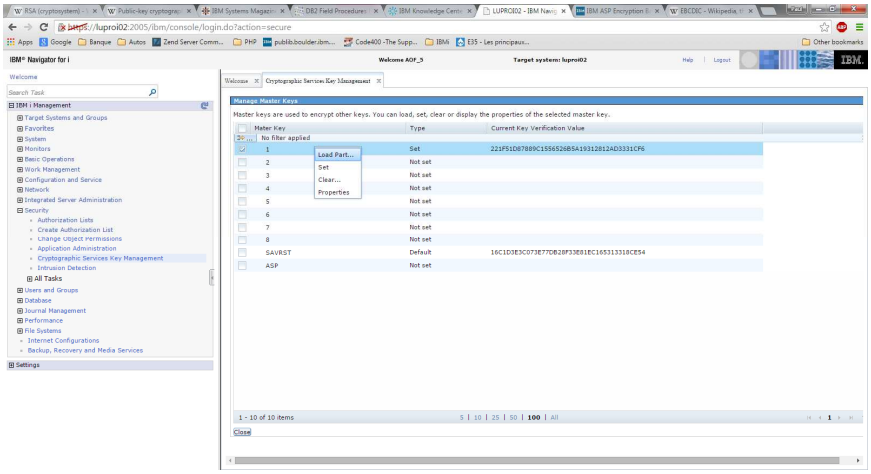
- select DECRYPT_CHAR(clientname, 'CommonDemo'), CARDNBR from common/credcard3

Cryptographic Services Key Management

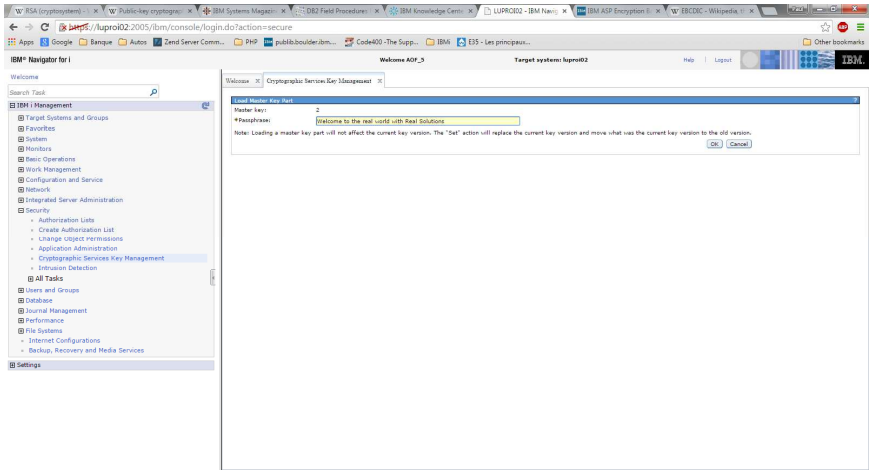
- HTTP://IBMiServer:2001



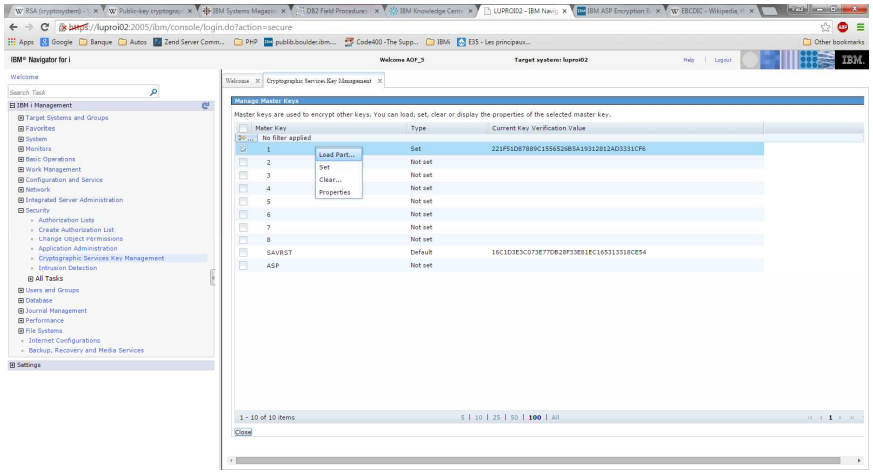
LOAD PARTS



Load Parts



SET KEY



ASP disk encryption

- install IBM i Option 45 (Encrypted ASP Enablement)
- Start SST
  - Work with disk Units
  - Work with disks configuration
  - Add units to ASP

SST

Add Units to ASPs

Select one of the following:

1. Create unencrypted ASPs
2. 2. Create encrypted ASPs
3.
4. 3. Add units to existing ASPs

Selection

F3=Exit F12=Cancel

SST

```
    Specify New Encrypted ASPs to Add Units to

Specify the new ASP to add each unit to.
All the new ASPs will be encrypted.
Specify Serial                              Resource
 ASP Number       Type Model Capacity      Name
____ 68-0D07090   4326 072    30769        DD008


F3=Exit F5=Refresh F11=Display disk configuration
capacity F12=Cancel

Then confirm… etc..
```

SST

- Work with disk units
- Display disk configuration
- Display encryption status

- Redbook SG247680.pdf (chapter 8)

Independent ASP

- To encrypt independent ASP
  - Use System i Navigator or IBM system Director navigator for i
  - If you use geographic mirroring please notice that

  The encryption key MUST be the same on all systems in the cluster.

TAPE Encryption

- Hardware encryption
  - Faster but more expensive
  - Native save/restore command
- Software encryption
  - Low cost but slow…
  - BRMS

Hardware encryption

- Need special hardware
  - Fiber channel attachement
  - Encryption capable tape drive TS1120 in a library TS3400, TS3500 or 3494
  - Or A fiber LTO4 drive in a library TS3100, TS3200, TS3310 or TS3500
  - Encryption capable media
- Features
  - Transparent Library Managed Encryption Feature
  - Encryption Key Managers (EKM) with IBM Java Runtime Environment (JRE™)
  - 5761-SS1 Option 34 Digital Certificate Manager (DCM) is required if EKM is on IBM i.
  - BRMS recommended (not mandatory)

Software encryption

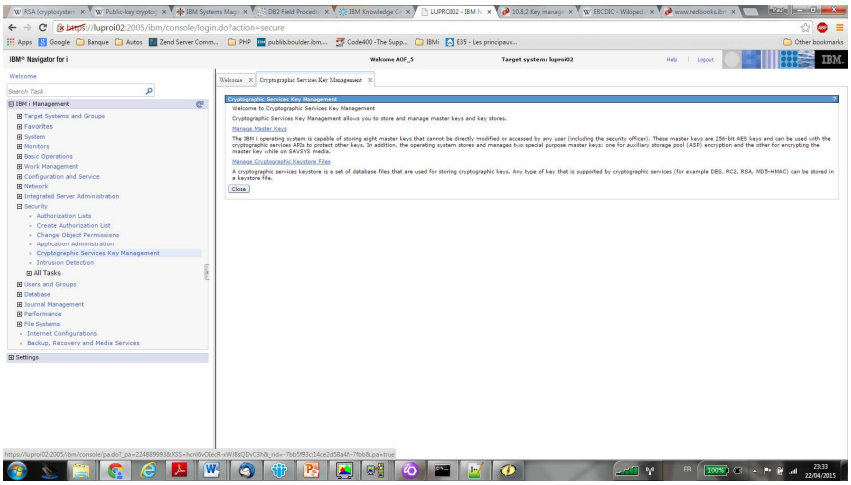
- Prerequisites
  - 5761-SS1 Option 18 - Media and Storage Extensions
  - 5761-SS1 Option 44 - Encrypted Backup enablement
  - 5761-BR1 Base - Backup Recovery and Media Services for i5/OS
  - 5761-BR1 Option 2 - BRMS-Advanced Functions Feature

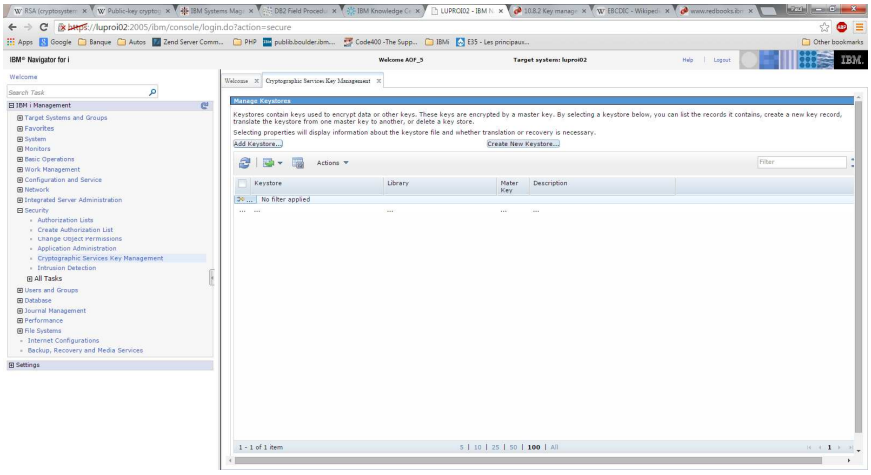
Software tape encryption

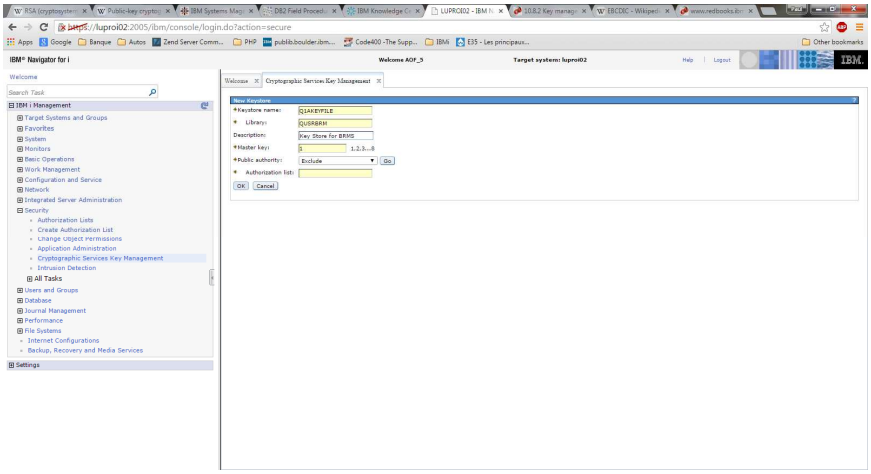
- Keystore file Q1AKEYFILE
- BRMS requires the only valid keystore file Q1AKEYFILE, and it must exist in library QUSRBRM

- If the keys are lost, the encrypted backup data on the save media cannot be restored.

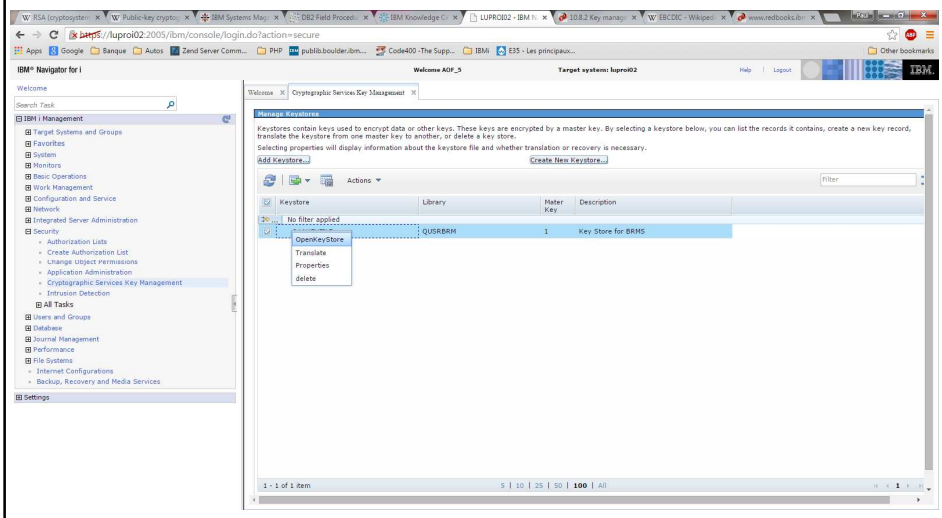
Creating the key store



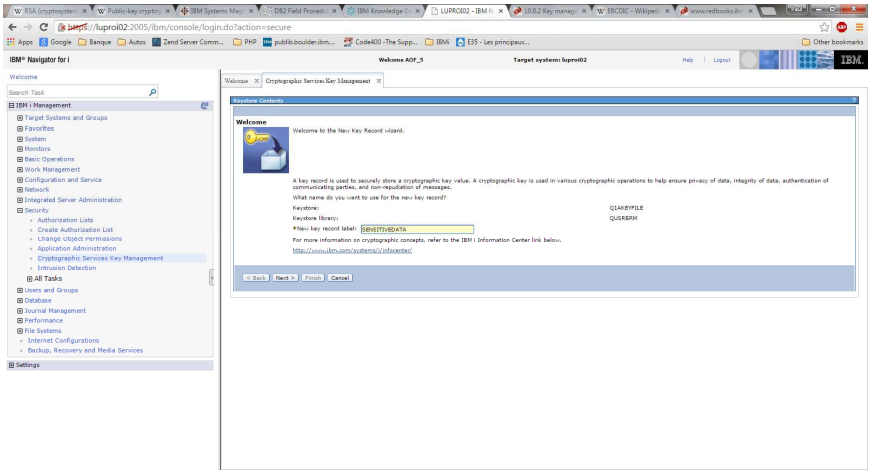

https://luproi02:2005/ibm/console/login.do?action=secure
IBM® Navigator for i
Welcome AOF_5
Target system: luproi02
Welcome
Cryptographic Services Key Management
Manage Keystores
Keystores contain keys used to encrypt data or other keys. These keys are encrypted by a master key. By selecting a keystore below, you can list the records it contains, create a new key record, translate the keystore from one master key to another, or delete a key store.
Selecting properties will display information about the keystore file and whether translation or recovery is necessary.
Add Keystore...
Create New Keystore...
Actions
Keystore
Library
Mster Key
Description
No filter applied
1 - 1 of 1 item
IBM i Management
Target Systems and Groups
Favorites
System
Monitors
Basic Operations
Work Management
Configuration and Service
Network
Integrated Server Administration
Security
Authorization Lists
Create Authorization List
Change Object Permissions
Application Administration
Cryptographic Services Key Management
Intrusion Detection
All Tasks
Users and Groups
Database
Journal Management
Performance
File Systems
Internet Configurations
Backup, Recovery and Media Services
Settings


https://luproi02:2005/ibm/console/login.do?action=secure
IBM® Navigator for i
Welcome AOF_5
Target system: luproi02
Welcome
Cryptographic Services Key Management
New Keystore
Keystore name:
QIAKEYFILE
Library:
QUSRBRM
Description:
Key Store for BRMS
Master key:
1
1,2,3...8
Public authority:
Exclude
Go
Authorization list:
OK
Cancel
IBM i Management
Target Systems and Groups
Favorites
System
Monitors
Basic Operations
Work Management
Configuration and Service
Network
Integrated Server Administration
Security
Authorization Lists
Create Authorization List
Change Object Permissions
Application Administration
Cryptographic Services Key Management
Intrusion Detection
All Tasks
Users and Groups
Database
Journal Management
Performance
File Systems
Internet Configurations
Backup, Recovery and Media Services
Settings

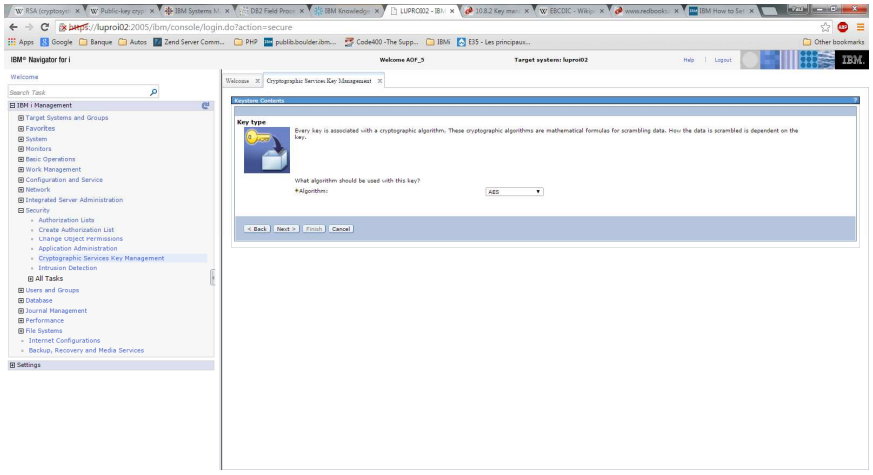
Creating the keys



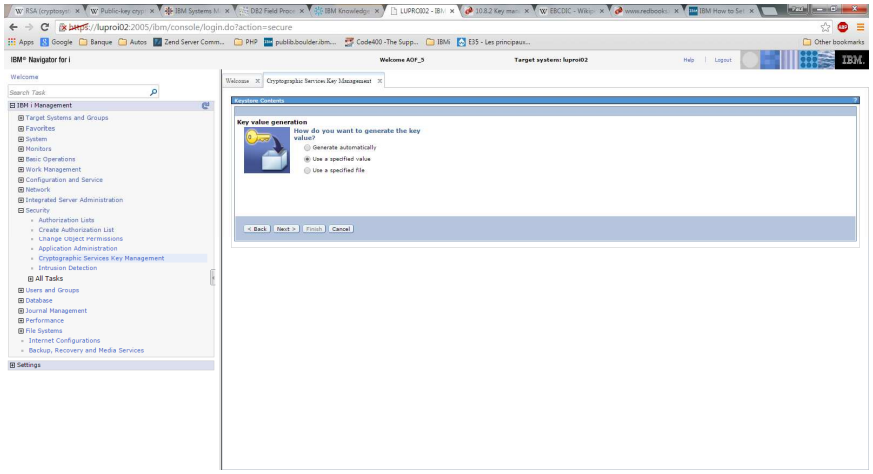
Entering the label



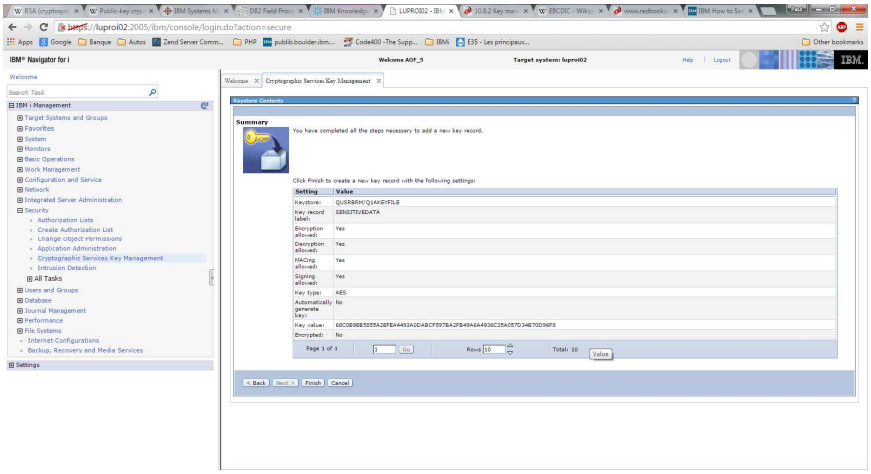
Specify algorithm AES



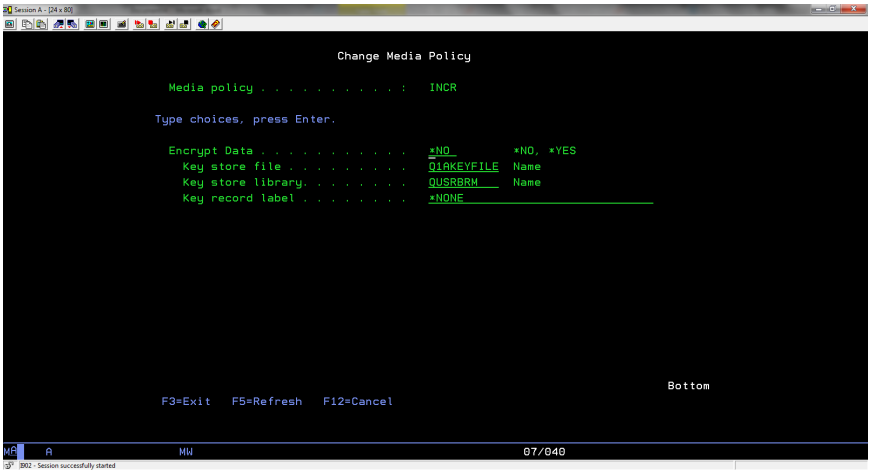
Generation value



Click finish



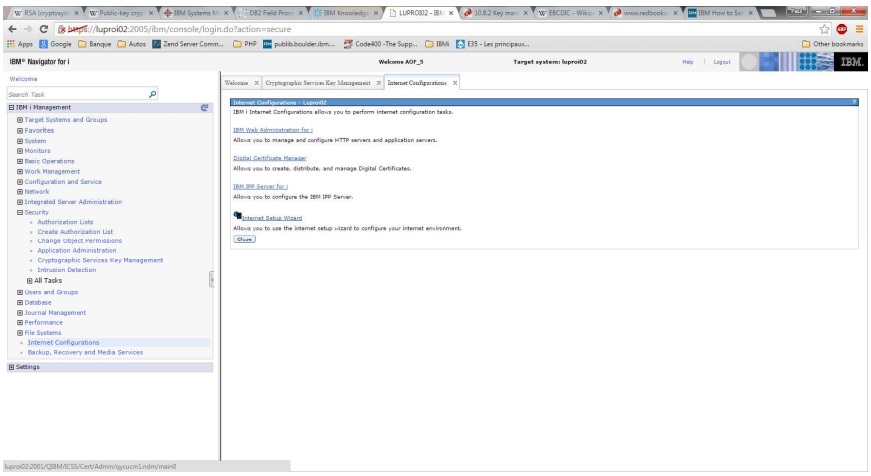
IN BRMS Media policy

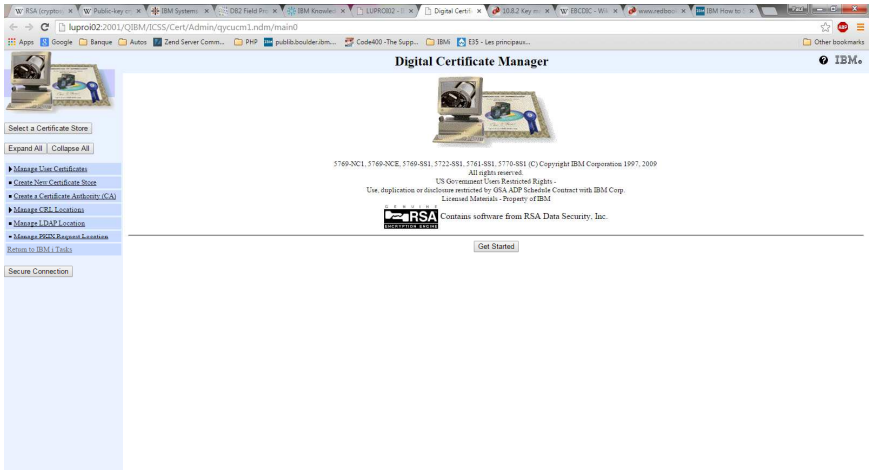


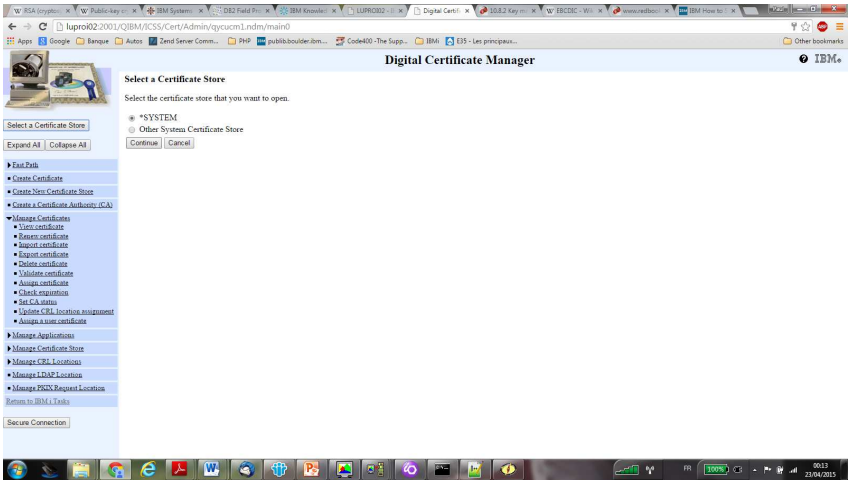
SSL

- Encryption of Communication
- Requires certificates
- Certificates are managed by DCM
- http://IBMiServer:2001

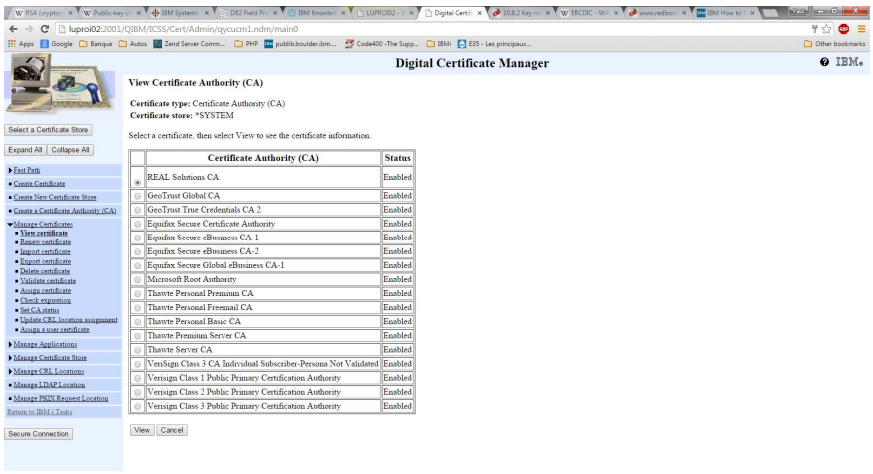
DCM




Digital Certificate Manager
IBM
Select a Certificate Store
Expand All
Collapse All
Manage User Certificates
Create New Certificate Store
Create a Certificate Authority (CA)
Manage CRL Locations
Manage LDAP Location
Manage PKIX Request Location
Return to IBM i Tasks
Secure Connection
5769-NC1, 5769-NCE, 5769-SS1, 5722-SS1, 5761-SS1, 5770-SS1 (C) Copyright IBM Corporation 1997, 2009
All rights reserved.
US Government Users Restricted Rights -
Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Licensed Materials - Property of IBM
Contains software from RSA Data Security, Inc.
Get Started


Digital Certificate Manager
IBM
Select a Certificate Store
Select the certificate store that you want to open.
*SYSTEM
Other System Certificate Store
Continue
Cancel
Select a Certificate Store
Expand All
Collapse All
Fast Path
Create Certificate
Create New Certificate Store
Create a Certificate Authority (CA)
Manage Certificates
View certificate
Renew certificate
Import certificate
Export certificate
Delete certificate
Validate certificate
Assign certificate
Check expiration
Set CA status
Update CRL location assignment
Assign a user certificate
Manage Applications
Manage Certificate Store
Manage CRL Locations
Manage LDAP Location
Manage PKIX Request Location
Return to IBM i Tasks
Secure Connection

CA ..



Document to configure SSL Telnet

- http://w ww-01.ibm.com/support/docview.wss?uid=nas8N1010449

sFTP

- Secure FTP is provided under PASE
  - Need PASE (option 33 of OS)
  - Need 5733-SC1 (Portable Utilities for i) contains OpenSSH
  - Call QP2TERM (PASE terminal)
  - chmod 755
  - ssh-keygen -t dsa -N "“ (generates the keys)
  - Send your public key to your partner
  - ssh-keyscan -t commpartner.com >> ~/.ssh/known_hosts (add your partner key to your host file)

sFTP

- Run in batch PGM sample
- Script.txt

```
put /settle/workfile.txt /incoming/workfile.txt
exit
```

- PGM

```
DCL    VAR(&RC) TYPE(*INT) LEN(4)
DCL    VAR(&MSGDTA) TYPE(*CHAR) LEN(4)
DCL    VAR(&MSGID) TYPE(*CHAR) LEN(7)
QSH CMD('/QOpenSys/usr/bin/sftp -vvv -b /Script/put_script.txt
   RMTUSER@commpartner.com > /logs/put_logoutput.txt 2>&1')
RCVMSG MSGTYPE(*COMP) RMV(*NO) MSGDTA(&MSGDTA) +
   MSGID(&MSGID)
CHGVAR VAR&RC) VALUE(0)
IF     COND(&MSGID *EQ 'QSH0005') THEN(CHGVAR +
   VAR(&RC) VALUE(%BIN(&MSGDTA)))
IF COND(&RC *NE 0) THEN(DO)
  /* Error processing */
ENDDO
```

- http://www.ibmsystemsmag.com/ibmi/administrator/systemsmanagement/sFTP-Tips/?page=1

sFTP server

- Is provided under SSH
- The sshd daemon must be started
- Need the key to be generated the first time
- ssh-keygen -t rsa1 -f /QOpenSys/QIBM/UserData/SC1/OpenSSH/openssh-4.7p1/etc/ssh_host_key -N ""
- ssh-keygen -t dsa -f /QOpenSys/QIBM/UserData/SC1/OpenSSH/openssh-4.7p1/etc/ssh_host_dsa_key -N ""
- ssh-keygen -t rsa -f /QOpenSys/QIBM/UserData/SC1/OpenSSH/openssh-4.7p1/etc/ssh_host_rsa_key -N ""

- RUN
  - QSH CMD ('/QOpenSys/usr/sbin/sshd')
  - CALL PGM(QP2SHELL) PARM('/QopenSys/usr/sbin/sshd')

- OR (From IBM i 6.1) STRTCPSVR *SSHD

- http://www.redbooks.ibm.com/redpapers/pdfs/redp4163.pdf
- http://wiki.midrange.com/index.php/SSH