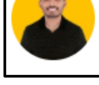
ENCRYPTION & DECRYPTION IN ABAP

SAVE FOR LATER



Swapnil More

SAP ABAP Developer @ SVKM | S/4HANA | OO-ABAP

Explaination of Encryption :

AES Encryption in ABAP

Encryption is like locking a box with a secret key.
Only the person who has the right **key** can open (decrypt) it.
Here we are using **AES-256 Algorithm** (industry standard, very secure).

Step by Step

Plain Text String
This is our normal string (plain text).

```
lv_plain_str = | Swapnil More - SAP ABAP Developer, Email - erswapnilmore@gmail.com |.
```

Convert String → UTF-8 XSTRING
ABAP encryption works only with XSTRING (binary format).
So we first convert our plain text to UTF-8 xstring.

```
lv_plain_x = cl_abap_codepage=>convert_to( source = lv_plain_str ).
```

Create a Key
For AES-256, the key must be 32 characters.
Here we created a sample key string and converted it into xstring.

```
lv_key_x = cl_bcs_convert=>string_to_xstring(
  iv_codepage = 'UTF-8'
  iv_string   = 'swapnilmoresapabapdeveloper1234 ‘ ).
```

Create an IV (Initialization Vector)
IV is like a salt / randomizer used along with the key to make encryption stronger.
For AES, it must be 16 characters.
Here we just used "0000000000000000" for demo, but in real cases it should be random.

```
lv_iv_x = cl_bcs_convert=>string_to_xstring(
  iv_codepage = 'UTF-8'
  iv_string   = '0000000000000000' ).
```

Swapnil More
SAP ABAP Developer @
SVKM | S/4HANA | OO-ABAP

Encrypt the Data

```
cl_sec_sxml_writer=>encrypt_iv(
  EXPORTING
    plaintext  = lv_plain_x
    key        = lv_key_x
    iv         = lv_iv_x
    algorithm  = cl_sec_sxml_writer=>co_aes256_algorithm_pem
  IMPORTING
    ciphertext = lv_cipher_x ).
```

This function locks your plain text with the key + IV.
Input: lv_plain_x (plain text in xstring)
Output: lv_cipher_x (cipher text in xstring, unreadable secret code)
Now your data is safe! 🚀

Convert Cipher → Base64

```
CALL FUNCTION 'SCMS_BASE64_ENCODE_STR'
  EXPORTING
    input  = lv_cipher_x
  IMPORTING
    output = lv_cipher_b64.
```

XSTRING is still not human-friendly.
So we convert cipher into Base64 string, which looks like:
D7+AYuj2x1qz7E3Pk0X1...
Handle Special Characters (Optional)

```
*REPLACE ALL OCCURRENCES OF '+' IN lv_cipher_b64 WITH '-'.
*REPLACE ALL OCCURRENCES OF '/' IN lv_cipher_b64 WITH '_'.
*REPLACE ALL OCCURRENCES OF '=' IN lv_cipher_b64 WITH ‘’.
```

Some systems (like URLs, JSON APIs) don’t like special characters (+ / =).
So we replace them with safe characters.
Final Result:
Your simple readable text → converted into encrypted Base64 string.
Only someone with the same Key + IV can unlock it again.

Swapnil More
SAP ABAP Developer @ SVKM | S/4HANA | OO-ABAP

AES Decryption in ABAP

Decryption is just the reverse process of Encryption.
We take the encrypted Base64 string, unlock it with the same Key + IV, and get back the original plain text.

Convert Base64 → XSTRING

```
CALL FUNCTION 'SCMS_BASE64_DECODE_STR'
  EXPORTING
    input  = lv_cipher_b64
  IMPORTING
    output = lv_cipher_x.
```

Our encrypted data was stored in Base64 format (easy to transport).
Now we decode it back to the original cipher xstring.

Decrypt with Key + IV

```
cl_sec_sxml_writer=>decrypt(
  EXPORTING
    ciphertext = lv_cipher_x
    key        = lv_key_x
    algorithm  = cl_sec_sxml_writer=>co_aes256_algorithm_pem
  IMPORTING
    plaintext  = lv_decrypted_x ).
```

Now the magic happens!
Input: lv_cipher_x (encrypted xstring)
Key: lv_key_x (same key we used in encryption)
Output: lv_decrypted_x (plain xstring, still not human-readable).
If the Key or IV is wrong → you won't get the correct data.

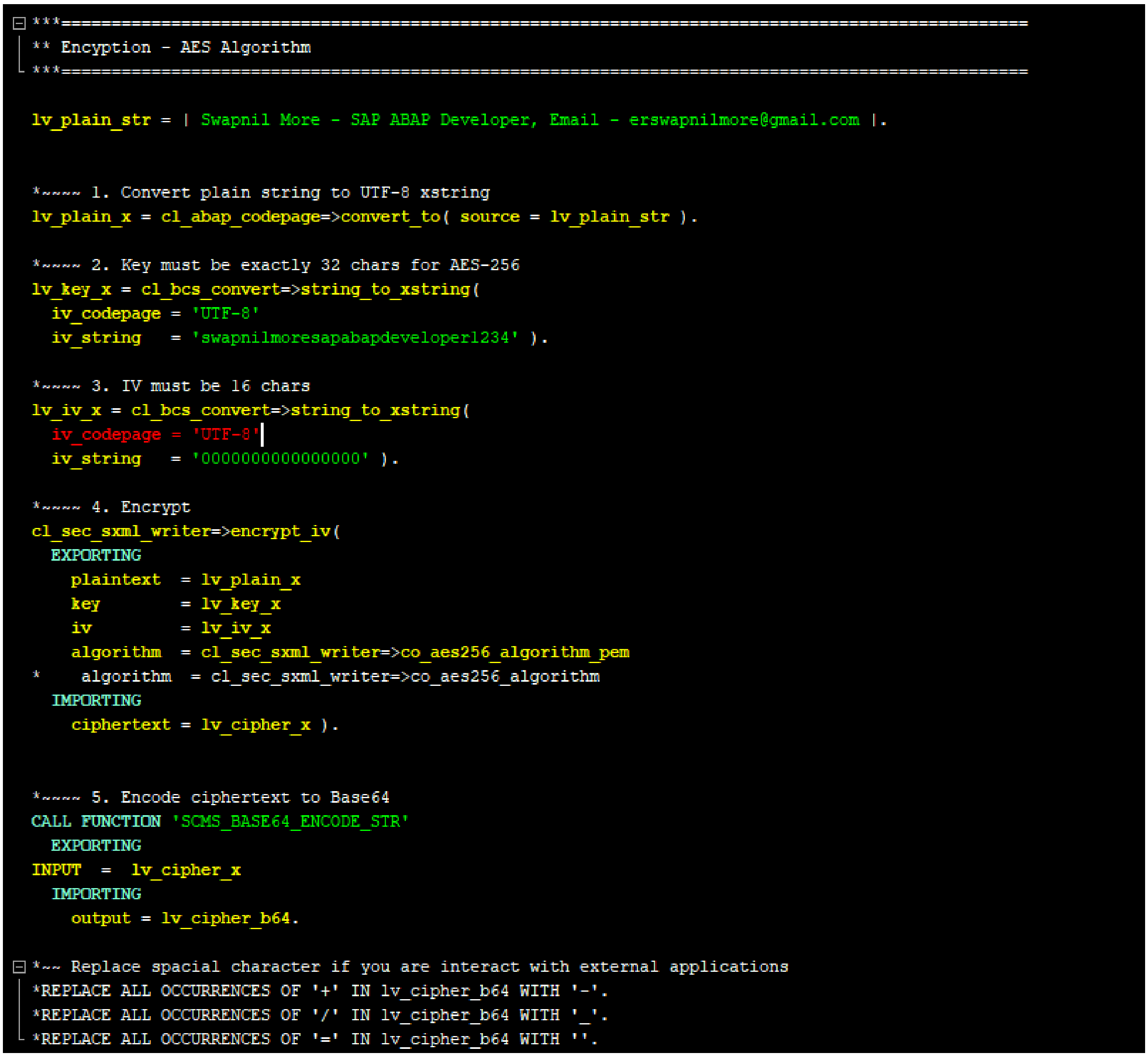
Convert Decrypted XSTRING → String

```
cl_abap_conv_in_ce=>create( input = lv_decrypted_x )->read(
  IMPORTING data = lv_decrypted_str ).
```

Finally, we change the decrypted xstring back into a normal readable string.
So lv_decrypted_str will give us:

```
Swapnil More - SAP ABAP Developer, Email - erswapnilmore@gmail.com
Success! 🎉
```

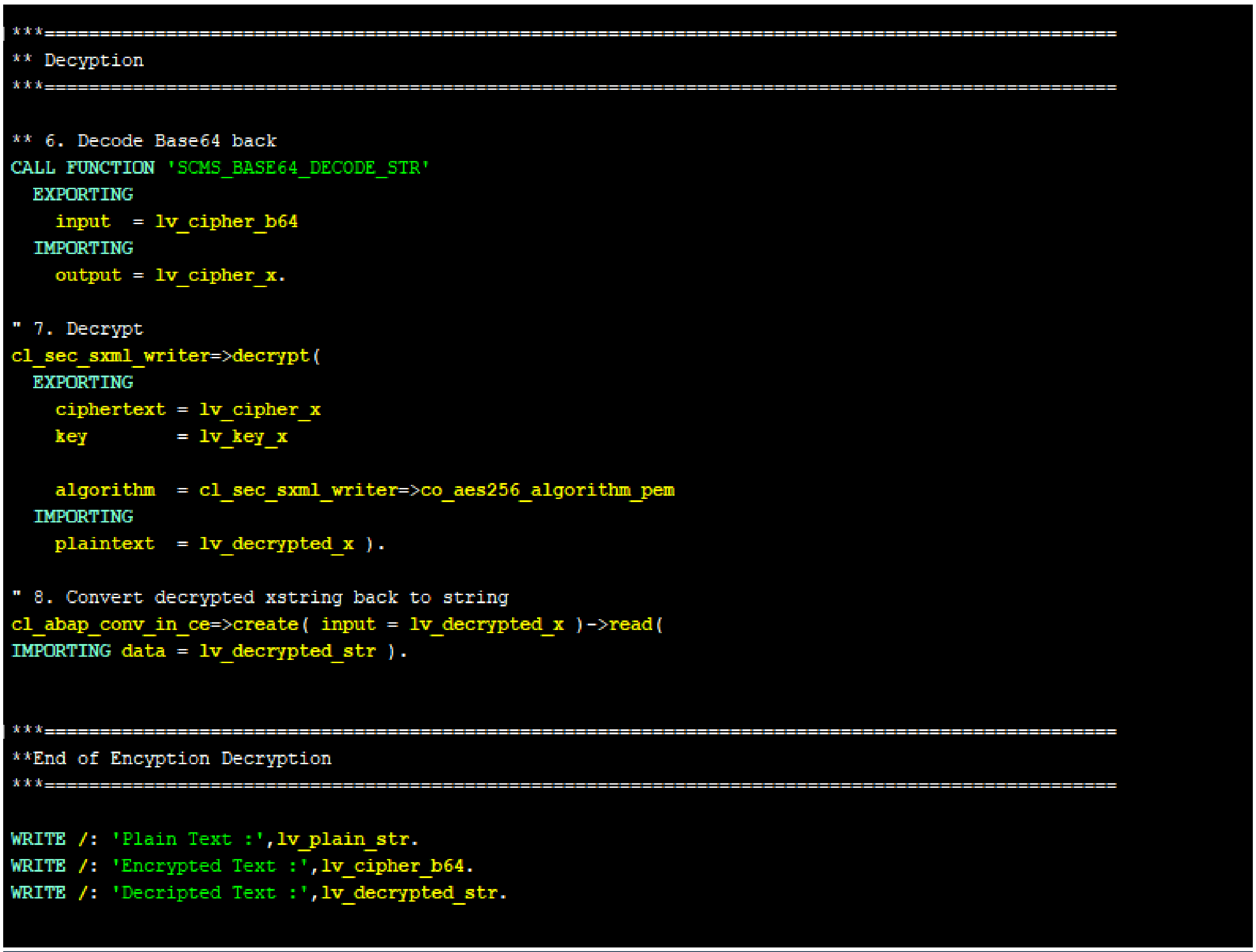
Swapnil More
SAP ABAP Developer @
SVKM | S/4HANA | OO-ABAP



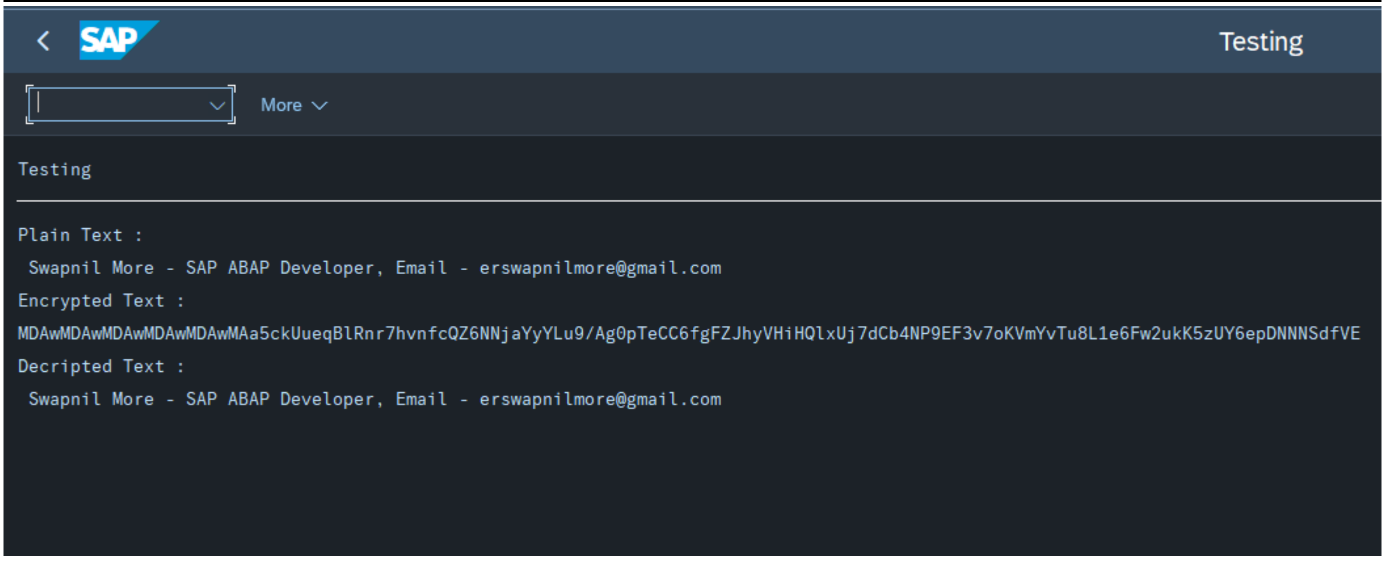
```
***=================================================================================================
** Encyption - AES Algorithm
***=================================================================================================

lv_plain_str = | Swapnil More - SAP ABAP Developer, Email - erswapnilmore@gmail.com |.


*~~~~ 1. Convert plain string to UTF-8 xstring
lv_plain_x = cl_abap_codepage=>convert_to( source = lv_plain_str ).

*~~~~ 2. Key must be exactly 32 chars for AES-256
lv_key_x = cl_bcs_convert=>string_to_xstring(
  iv_codepage = 'UTF-8'
  iv_string   = 'swapnilmoresapabapdeveloper1234' ).

*~~~~ 3. IV must be 16 chars
lv_iv_x = cl_bcs_convert=>string_to_xstring(
  iv_codepage = 'UTF-8'
  iv_string   = '0000000000000000' ).

*~~~~ 4. Encrypt
cl_sec_sxml_writer=>encrypt_iv(
  EXPORTING
    plaintext  = lv_plain_x
    key        = lv_key_x
    iv         = lv_iv_x
    algorithm  = cl_sec_sxml_writer=>co_aes256_algorithm_pem
*    algorithm  = cl_sec_sxml_writer=>co_aes256_algorithm
  IMPORTING
    ciphertext = lv_cipher_x ).


*~~~~ 5. Encode ciphertext to Base64
CALL FUNCTION 'SCMS_BASE64_ENCODE_STR'
  EXPORTING
INPUT   =  lv_cipher_x
  IMPORTING
    output = lv_cipher_b64.

*~~ Replace spacial character if you are interact with external applications
*REPLACE ALL OCCURRENCES OF '+' IN lv_cipher_b64 WITH '-'.
*REPLACE ALL OCCURRENCES OF '/' IN lv_cipher_b64 WITH '_'.
*REPLACE ALL OCCURRENCES OF '=' IN lv_cipher_b64 WITH ''.
```

Swapnil More

SAP ABAP Developer @
SVKM | S/4HANA | OO-ABAP



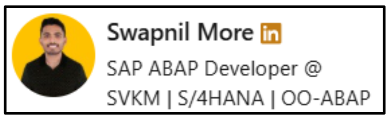
```
***=================================================================================================
** Decyption
***=================================================================================================

** 6. Decode Base64 back
CALL FUNCTION 'SCMS_BASE64_DECODE_STR'
  EXPORTING
    input  = lv_cipher_b64
  IMPORTING
    output = lv_cipher_x.

" 7. Decrypt
cl_sec_sxml_writer=>decrypt(
  EXPORTING
    ciphertext = lv_cipher_x
    key        = lv_key_x

    algorithm  = cl_sec_sxml_writer=>co_aes256_algorithm_pem
  IMPORTING
    plaintext  = lv_decrypted_x ).

" 8. Convert decrypted xstring back to string
cl_abap_conv_in_ce=>create( input = lv_decrypted_x )->read(
IMPORTING data = lv_decrypted_str ).


***=================================================================================================
**End of Encyption Decryption
***=================================================================================================

WRITE /: 'Plain Text :',lv_plain_str.
WRITE /: 'Encrypted Text :',lv_cipher_b64.
WRITE /: 'Decripted Text :',lv_decrypted_str.
```



```
Testing

Plain Text :
 Swapnil More - SAP ABAP Developer, Email - erswapnilmore@gmail.com
Encrypted Text :
MDAwMDAwMDAwMDAwMDAwMAa5ckUueqBlRnr7hvnfcQZ6NNjaYyYLu9/Ag0pTeCC6fgFZJhyVHiHQlxUj7dCb4NP9EF3v7oKVmYvTu8L1e6Fw2ukK5zUY6epDNNNSdfVE
Decripted Text :
 Swapnil More - SAP ABAP Developer, Email - erswapnilmore@gmail.com
```

Swapnil More
SAP ABAP Developer @
SVKM | S/4HANA | OO-ABAP

Connect for more updates


Swapnil More
SAP ABAP Developer @ SVKM | S/4HANA | OO-ABAP

Click here

topmate.io/erswapnilmore/

Click here

SAP PRIME | WhatsApp Channel
SAP PRIME WhatsApp Channel. *SAP PRIME – Let's Crack it* 📢 • Daily QnA - Interview preparation with me • SA...

Click here