User Manual of
DUKPT (Derive Unique Key Per Transaction)
Testing Tool

User Manual

(Revision 1.1)

Contents

Figures

Tables

1. Revision History

Revision	Date	Author	Comments
1.0	05/09/2023	C. Chen	Initial document
1.1	05/14/2025	C. Chen	(1) Revised for the UI change in Sec 5.3. (2) Added examples in Chap 6.

2. Introduction

2.1 About the DUKPT

DUKPT (**D**erive **U**nique **K**ey **P**er **T**ransaction) is a key management algorithm which can generate a unique key whenever a transaction is processed. The advantages of the DUKPT are:

- The algorithm just needs a single **BDK** (**B**ase **D**erivation **K**ey) and one counter **KSN** (**K**ey **S**erial **N**umber) to derive all DUKPT keys for every transaction from each device.
- The KSN contains two portions:
  - The first portion (say, it is the **initial KSN**) is used to derive the **IPEK** (**I**nitial **P**IN **E**ncryption **K**ey).
  - Each device can be assigned with a unique Initial KSN, so the loaded IPEK in every device is also unique. This will ensure that every device will have the different IPEK and KSN, which can guarantee no duplicate DUKPT keys will be generated.
  - The second portion is a serial counter which will increase every time during every transaction. This makes the derived DUKPT session keys also be unique every time in every transaction.
- Whenever an encrypted transaction data and KSN are sent to the host, the host can easily derive the correct DUKPT session key based on the same BDK and the current KSN, then decrypt the transaction data.
- The derivation of each DUKPT key does *not* depend on the previous key which can avoid the possible synchronization issue.
- With the above features, it is almost impossible to have the same key generated on any devices for any transactions.

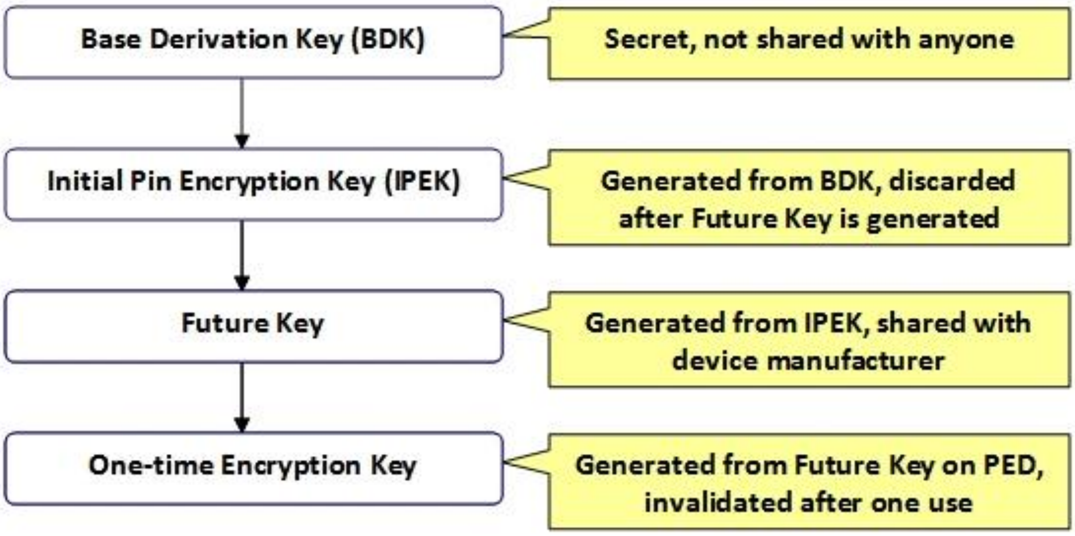

Base Derivation Key (BDK)
Secret, not shared with anyone
Initial Pin Encryption Key (IPEK)
Generated from BDK, discarded after Future Key is generated
Future Key
Generated from IPEK, shared with device manufacturer
One-time Encryption Key
Generated from Future Key on PED, invalidated after one use

Figure 1. The concept of DUKPT

2.2 Acronyms

Acronyms	Complete terminologies
BDK	Base Derivation Key
CBC	Cipher Block Chaining
CL	Contactlesss
CT	Contact
DES	Data Encryption Standard
DUKPT	Derive Unique Key Per Transaction
ECB	Electronic Code Book
HEX	Hexadecimal
ICV	Initial Chain Vector
IPEK	Initial PIN Encryption Key
KCV	Key Check Value
KSN	Kery Serial Number
MAC	Message Authentication Code
PAN	Primary Account Number
PIN	Personal Identification Number
PED	PIN Encryption Device
TDES	Triple DES
XOR	Exclusive-OR

3. Software Installation

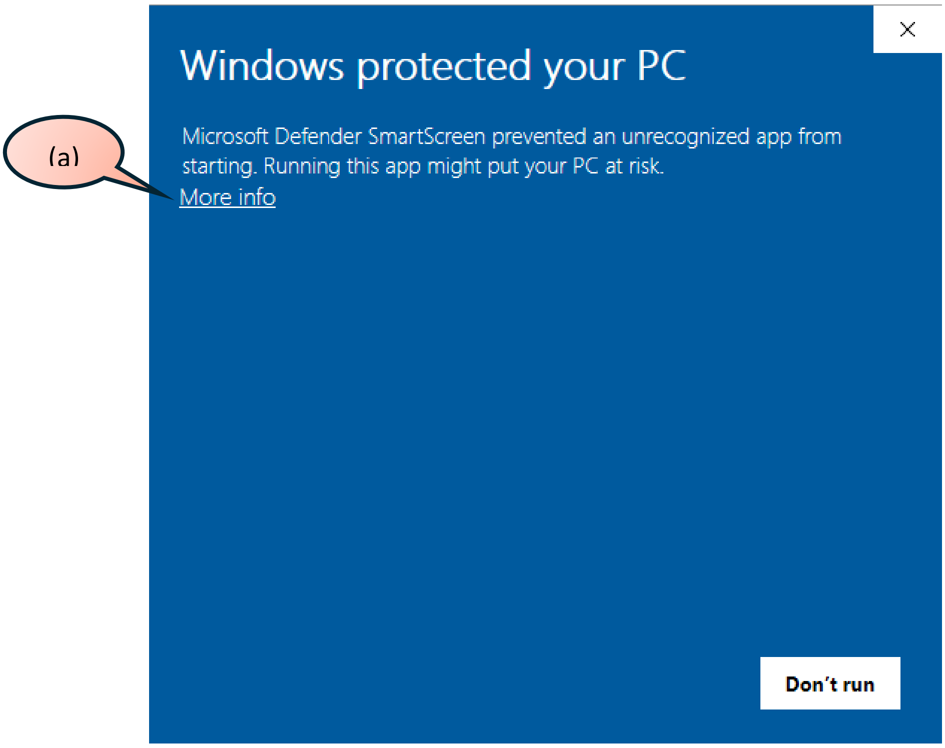
This DUKPT Testing Tool (hereinafter as the "**Tool**") can only be run under Windows environment. Windows 10, 11 or earlier version should be fine.

There is **no** installation required for running this Tool. You just need to copy all files (as listed in the Table 1 below) to the same folder, and then double click the **DUKPT.exe** to run.

Filenames	Description
DULPT.exe	The main program for execution
DUKPT_Data.xml	The DUKPT key data of all acquirers. Whenever there is a new Acquirer and/or DUKPT keys, they should be added to this file.
Proj_DES_DLL.dll	Dynamic library
borlndmm.dll	Dynamic library
cc32c270mt.dll	Dynamic library
rtl270.bpl	Dynamic library
vcl270.bpl	Dynamic library

Table 1 – Required files for running DUKPT Tool

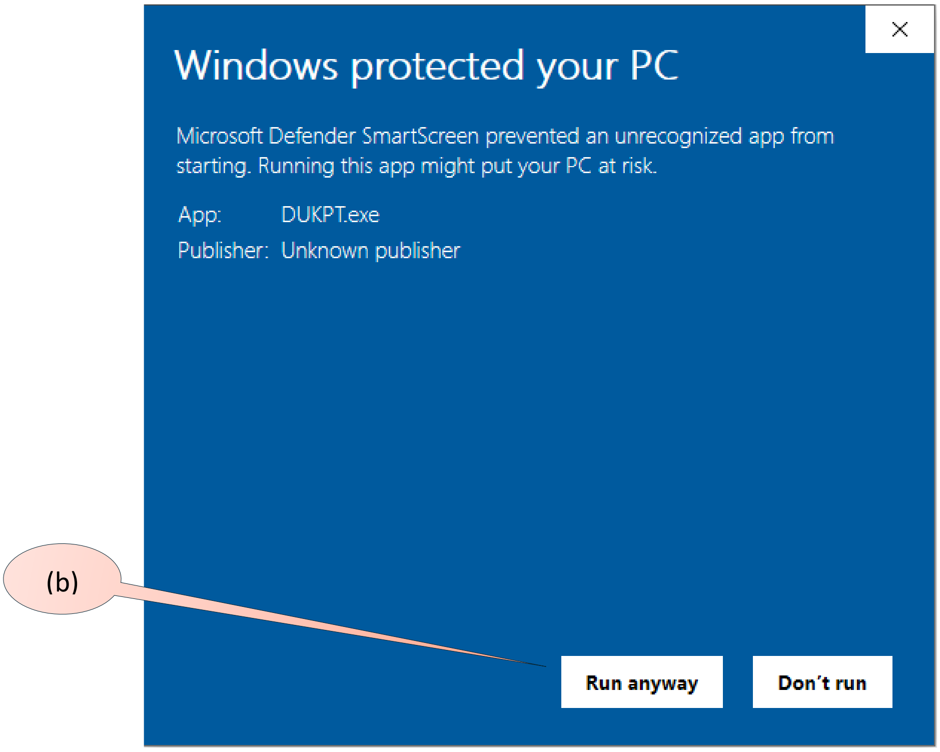
However, when you run the Tool for the first time, Windows might find it is a brand-new program and will prompt the warning message as Figure 2 shows. Please click (a) More info and (b) "Run anyway" button (in Figure 3) to skip the warning message. Normally, this would occur once only.



Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

Figure 2. Warning of running DUKPT Tool for the first time



Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

App: DUKPT.exe

Publisher: Unknown publisher

Figure 3. Enable to run DUKPT Tool for the first time

Then, you can start to use the Tool (see Figure 5).

4. Parameter File

4.1 Architecture of BDKs

In GoDaddy's key management architecture, the DUKPT algorithm might be used for

(a) data encryption,
(b) PIN encryption, and/or
(c) MAC (*Message Authentication Code*) calculation.

For the acquirers that GoDaddy works with, there might be up to 3 BDKs specified in the payment applications. All the BDKs of each acquirer are defined in a file of ***DUKPT_Data.xml.*** BDKs and KSNs are mandatory for this Tool, if you know there are new BDKs, please help to add them to the ***DUKPT_Data.xml.***

When you select one acquirer, the Tool will use the specified data in the parameter file to retrieve the BDKs and KSNs for later calculation. If we use Elavon CA as an example, the data architecture is specified like:

```
<Acquirer>
    <Name type="ACQ" value="Elavon_CA" />
    <DUKPT_PAN>
        <Name type="BDK" value="748D72BEB4DD5027BD144E6090953A99" />
        <Name type="KCV" value="A3AC03F88CA6C5D8" />
        <Name type="KSN" value="FFFF998890" />
    </DUKPT_PAN>
    <DUKPT_PIN>
        <Name type="BDK" value="81D8A6E10F1244F97D48D25A772E565A" />
        <Name type="KCV" value="919FE802D8CEFF47" />
        <Name type="KSN" value="FFFF998889" />
     </DUKPT_PIN>
     <DUKPT_MAC>
        <Name type="BDK" value="001E3F1E110C59B130C6C3B17E5C1DB1" />
        <Name type="KCV" value="DD9F6C79F6D1DE19" />
        <Name type="KSN" value="FFFF998891" />
    </DUKPT_MAC>
</Acquirer>
```

❖ All the BDKs of an acquirer are defined in the envelop of *<Acquirer>...</Acquirer>*

❖ The first line in the envelop is to define the Acquirer's name, which is "**Elavon CA**" in the example.

❖ There are 3 BDKs used by Elavon CA, and each BDK is defined in

- *<DUKPT_PAN>...</DUKPT_PAN>* ➔ for sensitive data encryption (e.g., PAN, Track 1, Track 2)
- *<DUKPT_PIN>...</DUKPT_PIN>* ➔ for online PIN block encryption
- *<DUKPT_PAN>...</DUKPT_MAC>* ➔ for Interac MAC calculation

❖ Each DUKPT envelop contains 3 components:

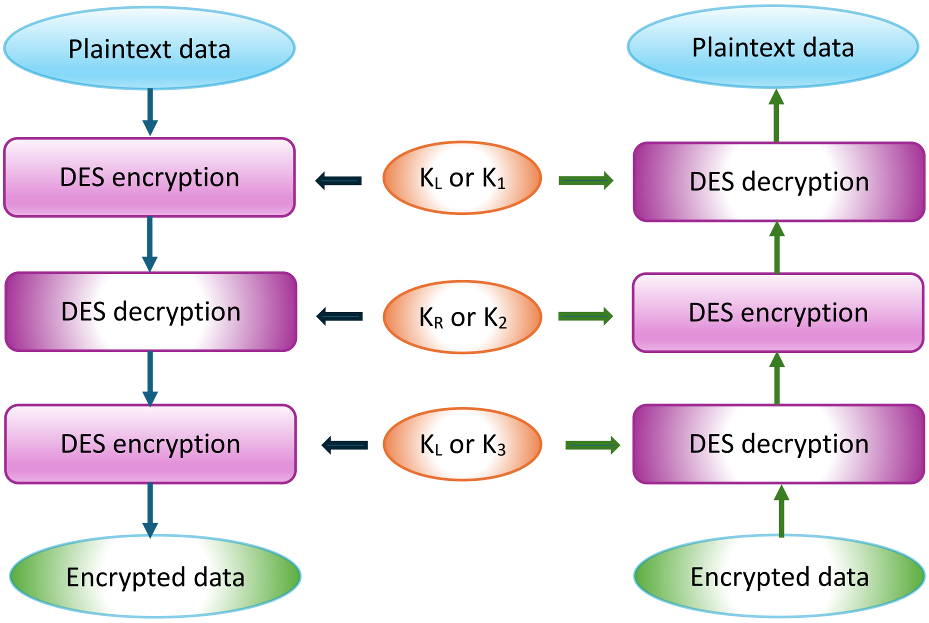
- *BDK* ➔ the data value of the BDK
- *KCV* ➔ the check value of BDK (optional)
- *KSN* ➔ the initial value of KSN for deriving IPEK (for reference)

So, when you select the acquirer and the type of BDK in the Tool, you will see the expected BDK and KSN shown on the screen (see Figure 5).

4.2 About the Key Length

To perform the Triple DES (TDES) calculation (see Figure 4), the key length can be double (16 byes, 2 parts – K_L || K_R) or triple length (24 bytes, 3 parts – K_1 || K_2 || K_3). GoDaddy is used to define the key in triple length. But, if you take a careful look of the key parts, the third part (K_3, Byte 17 ~ 24) is the same as the first part (K_1, Byte 1 ~ 8), which means the triple-length key is actually a double-length key ($K_1 = K_3$). So, it doesn't matter whether you fill in the double length BDK or triple length if the third part value is the same as the first part.

Besides, the DUKPT calculation was developed based on the double length key, which means the input and output key lengths are always 16 bytes. Therefore, the Tool does not extend the derived DUKPT session key to triple length.



Double length key (16 bytes): K_L || K_R
Triple length key (24 bytes): K_1 || K_2 || K_3

Figure 4. Concept of TDES calculation

5. Use the Tool

When the Tool is launched, there are 3 tabs for executing different functions.

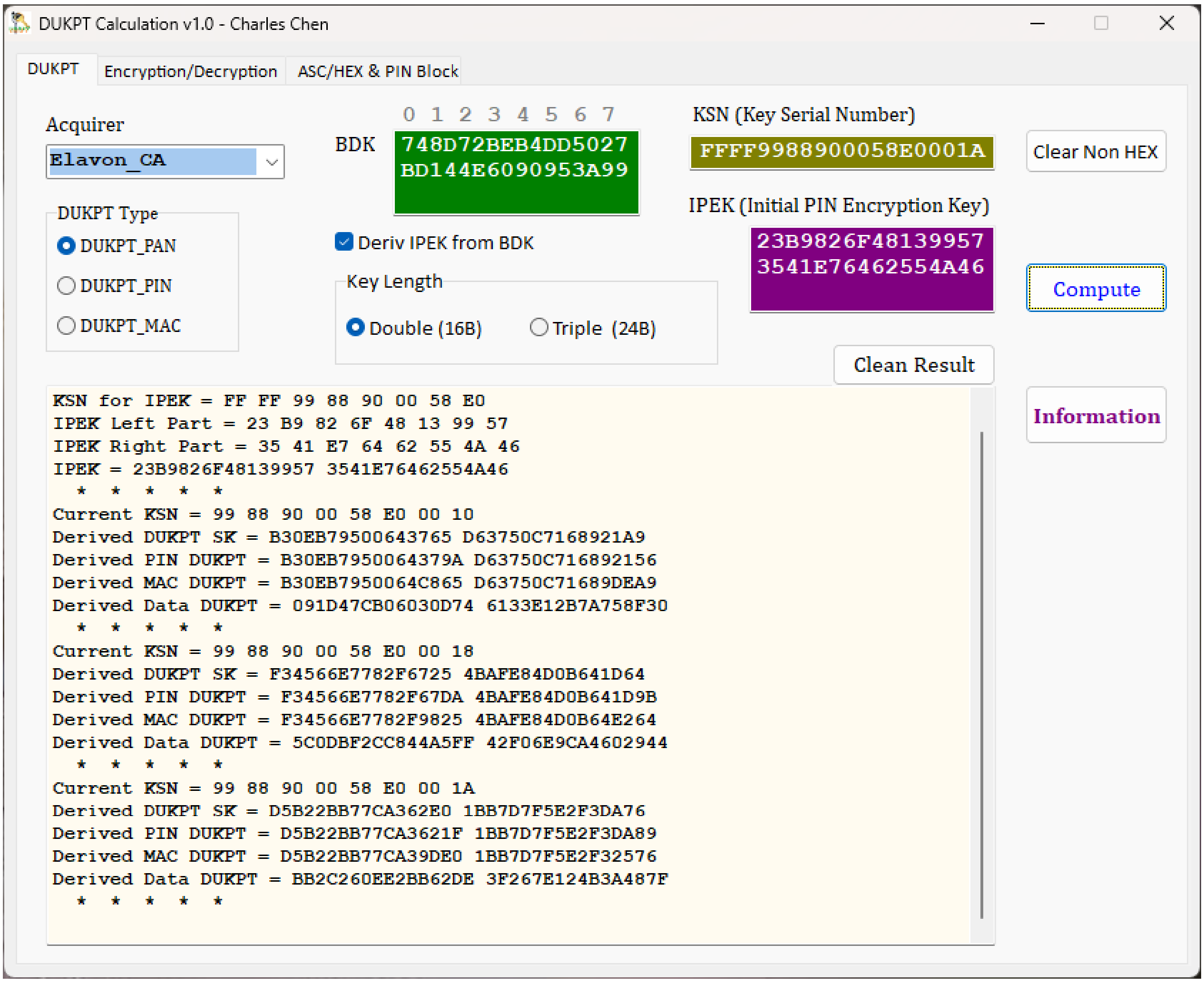
5.1 Tab “DUKPT” – DUKPT Session Key derivation

To derive the expected DUKPT session key with a given KSN, you should:

(1) If the BDK has been specified in the ***DUKPT_Data.xml***, you can select the acquirer from the Acquirer pull-down menu. In the meantime, a radiobutton group DUKPT Type will be shown.

- If there is *no* specific acquirer to select but you know the BDK, you can copy the BDK data to the BDK field directly.

(2) Next, based on your task requirement, select the BDK (PAN, PIN, or MAC) that you want to use for deriving the IPEK and DUKPT session key from the DUKPT Type. If the BDK and KSN are specified in the parameter file, they will be filled into the relative fields automatically.



Acquirer: Elavon_CA

BDK: 748D72BEB4DD5027 BD144E6090953A99

KSN (Key Serial Number): FFFF9988900058E0001A

IPEK (Initial PIN Encryption Key): 23B9826F48139957 3541E76462554A46

DUKPT Type: ● DUKPT_PAN ○ DUKPT_PIN ○ DUKPT_MAC

☑ Deriv IPEK from BDK

Key Length: ● Double (16B) ○ Triple (24B)

```
KSN for IPEK = FF FF 99 88 90 00 58 E0
IPEK Left Part = 23 B9 82 6F 48 13 99 57
IPEK Right Part = 35 41 E7 64 62 55 4A 46
IPEK = 23B9826F48139957 3541E76462554A46
   *   *   *   *   *
Current KSN = 99 88 90 00 58 E0 00 10
Derived DUKPT SK = B30EB79500643765 D63750C7168921A9
Derived PIN DUKPT = B30EB7950064379A D63750C716892156
Derived MAC DUKPT = B30EB7950064C865 D63750C71689DEA9
Derived Data DUKPT = 091D47CB06030D74 6133E12B7A758F30
   *   *   *   *   *
Current KSN = 99 88 90 00 58 E0 00 18
Derived DUKPT SK = F34566E7782F6725 4BAFE84D0B641D64
Derived PIN DUKPT = F34566E7782F67DA 4BAFE84D0B641D9B
Derived MAC DUKPT = F34566E7782F9825 4BAFE84D0B64E264
Derived Data DUKPT = 5C0DBF2CC844A5FF 42F06E9CA4602944
   *   *   *   *   *
Current KSN = 99 88 90 00 58 E0 00 1A
Derived DUKPT SK = D5B22BB77CA362E0 1BB7D7F5E2F3DA76
Derived PIN DUKPT = D5B22BB77CA3621F 1BB7D7F5E2F3DA89
Derived MAC DUKPT = D5B22BB77CA39DE0 1BB7D7F5E2F32576
Derived Data DUKPT = BB2C260EE2BB62DE 3F267E124B3A487F
   *   *   *   *   *
```

Figure 5. Main screen of DUKPT Testing Tool

(3) The pre-filled KSN value is for deriving IPEK. You should find and copy the KSN value to the KSN (Key Serial Number) field now, so you can derive the final DUKPT session key in one step. The KSN can be found in

- Tag '1F8101' : KSN for PIN
- Tag '1F8102' : KSN for data
- Tag '1F820B' : KSN for Interac Terminal generated MAC

(4) Once the BDK and the KSN (Key Serial Number) fields are filled in with the correct values, please click Compute button to start the IPEK and DUKPT session key derivation.

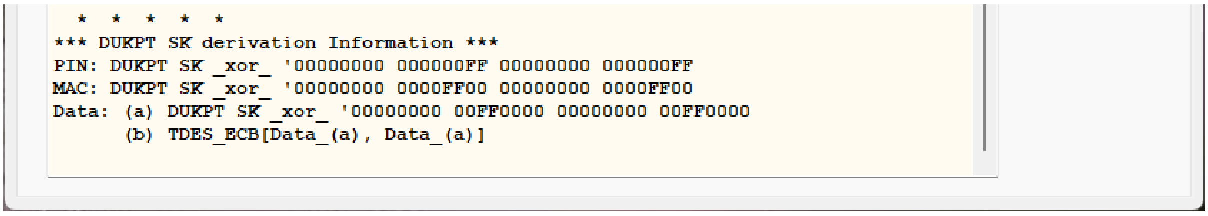
- The derived IPEK data will be filled into the IPEK field.
- The derivation of DUKPT session keys may need to be performed several times based on the value of KSN. The derivation results of every step will be shown in the information (light yellow) box below.

(5) Although the derivation results may show: "*PIN DUKPT*", "*MAC DUKPT*" and "*Data DUKPT*" in each derivation step, you should pick up the corresponding one only for your operation. GoDaddy key management uses different BDKs for different encryption types.

- For example, if you select the DUKPT_PIN in the DUKPT Type group, you should pick up the "*Derived PIN DUKPT*" from the most bottom for your PIN Block calculation.

Other functions in the page

- ❖ If you have IPEK rather than BDK data, you can uncheck Derive IPEK from BDK checkbox and fill the IPEK data into the IPEK field.
- ❖ You can click Clear Non HEX button to remove the non-Hexadecimal characters in all fields, if there are any.
- ❖ If you click Information button, it will tell how the button, it will tell how the *PIN DUKPT*, *MAC DUKPT* and *Data DUKPT* are calculated in the GoDaddy applications.

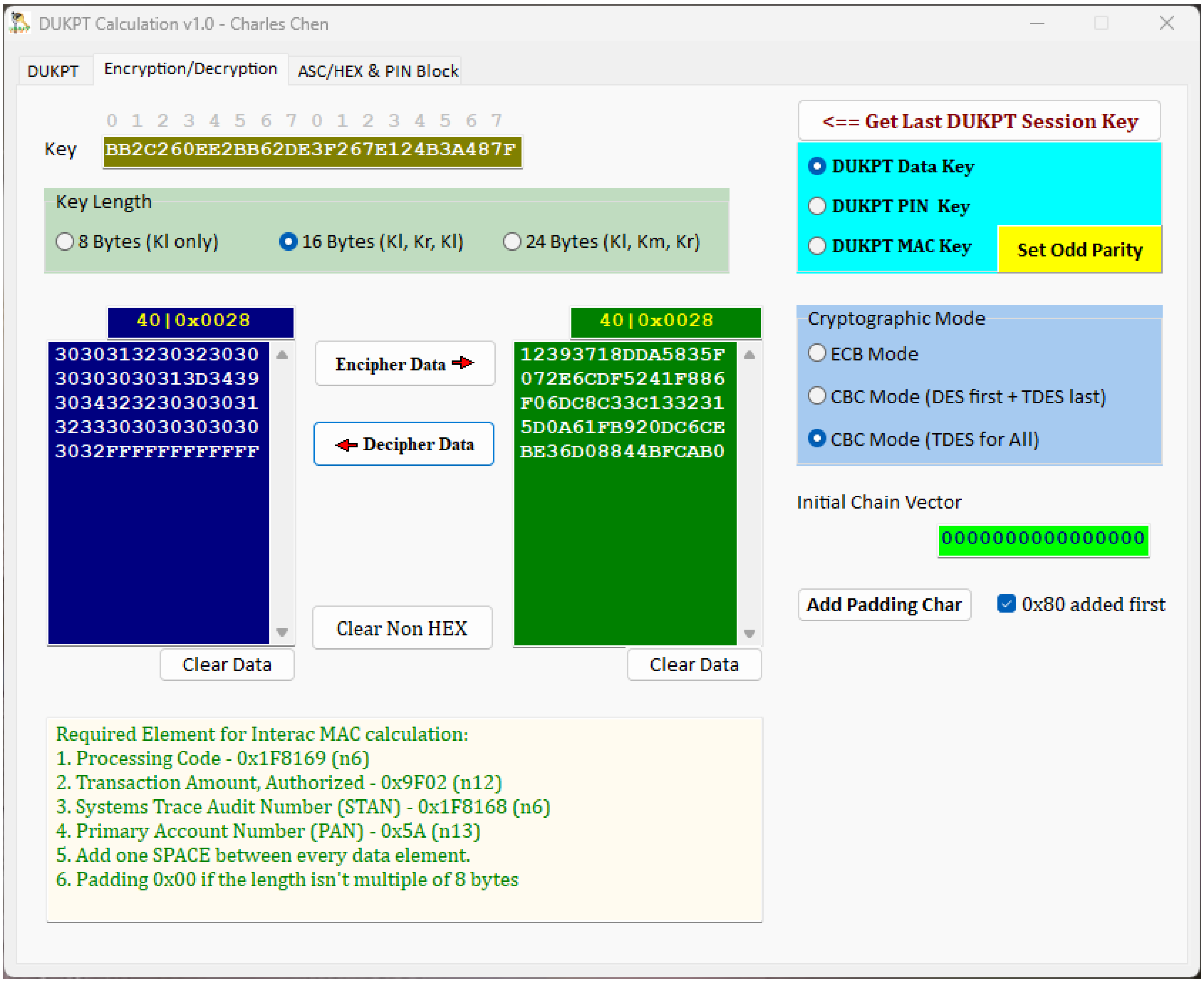


```
  *   *   *   *   *
*** DUKPT SK derivation Information ***
PIN: DUKPT SK _xor_ '00000000 000000FF 00000000 000000FF
MAC: DUKPT SK _xor_ '00000000 0000FF00 00000000 0000FF00
Data: (a) DUKPT SK _xor_ '00000000 00FF0000 00000000 00FF0000
      (b) TDES_ECB[Data_(a), Data_(a)]
```

Figure 6. Information about how PIN/MAC/Data DUKPT are calculated

5.2 Tab “Encryption/Decryption”

This tab is for TDES Encryption and Decryption, as shown below.



Key: BB2C260EE2BB62DE3F267E124B3A487F

Key Length: ○ 8 Bytes (Kl only) ● 16 Bytes (Kl, Kr, Kl) ○ 24 Bytes (Kl, Km, Kr)

● DUKPT Data Key
○ DUKPT PIN Key
○ DUKPT MAC Key

Cryptographic Mode:
○ ECB Mode
○ CBC Mode (DES first + TDES last)
● CBC Mode (TDES for All)

Initial Chain Vector: 0000000000000000

☑ 0x80 added first

40 | 0x0028

```
3030313230323030
30303030313D3439
3034323230303031
3233303030303030
3032FFFFFFFFFFFF
```

40 | 0x0028

```
12393718DDA5835F
072E6CDF5241F886
F06DC8C33C133231
5D0A61FB920DC6CE
BE36D08844BFCAB0
```

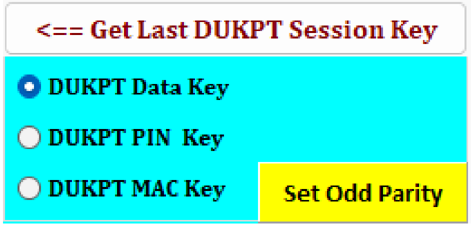
Required Element for Interac MAC calculation:
1. Processing Code - 0x1F8169 (n6)
2. Transaction Amount, Authorized - 0x9F02 (n12)
3. Systems Trace Audit Number (STAN) - 0x1F8168 (n6)
4. Primary Account Number (PAN) - 0x5A (n13)
5. Add one SPACE between every data element.
6. Padding 0x00 if the length isn't multiple of 8 bytes

Figure 7. Screen of encryption/decryption computation

Assign the key

You should fill in or select the key data in the Key field.

If the key was already derived in Sec. 5.1, you can easily copy the key by (a) selecting the key type; (b) and then clicking <== Get Last DUKPT Session Key button to copy the expected key data to the Key field.



● DUKPT Data Key
○ DUKPT PIN Key
○ DUKPT MAC Key

Figure 8. Select the key type and copy the key

When the key is copied, the Tool will judge and change the Key Length automatically.

Determine the Cryptographic Mode

There are 3 cryptographic modes that you should assign:

- **ECB (Electronic Coded Book) Mode** – Every data block (1 block = 8 bytes) is encrypted or decrypted independently.
- **CBC (Cipher Block Chaining) Mode (DES first + TDES last)** – If the data is separated into N blocks
  - the first to (N-1)th blocks are encrypted/decrypted with *single DES* CBC mode, then
  - the last Nth block is encrypted/decrypted with *triple DES* CBC mode.
  - For Interac MAC calculation, please select this.
- **CBC (Cipher Block Chaining) Mode (TDES for All)** – All the data blocks are encrypted/decrypted in *triple DES* CBC mode.
  - For PAN, Track 1/2 decryption, this mode shall be selected.

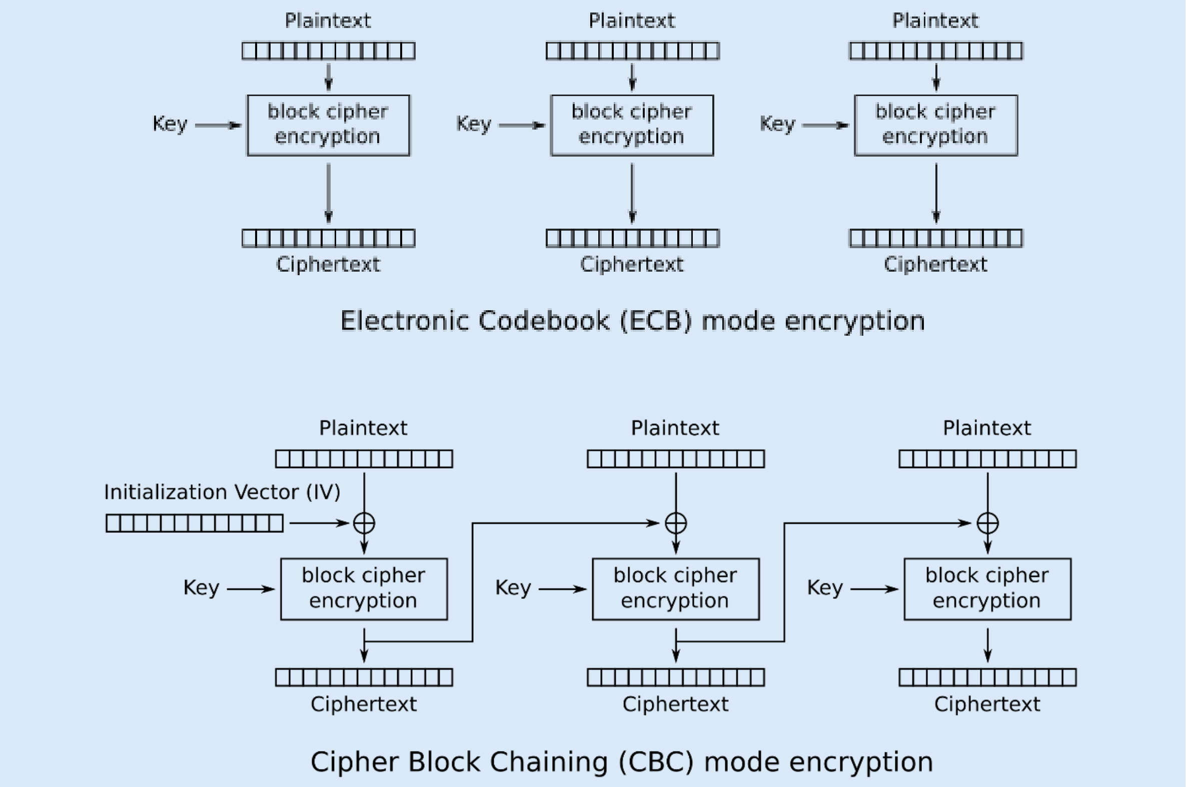


Figure 9. Diagrams about ECB and CBC modes

If the CBC mode is selected, another field of Initial Chain Vector (ICV) will pop out. Generally, the ICV value is set to 8-byte ‘00’. If not, you must fill in the correct value before doing the CBC calculation. For ECB mode, the ICV is not used, you will not see the Initial Chain Vector field.

Depending on the encryption/decryption requirements, you should select the proper mode to ensure the TDES calculation is correct.

Length of the Encryption Data

Due to the specification of DES, the length of the encryption data must be multiple of 8 bytes (8, 16, 24, ….). If the data length is not multiple of 8 bytes, it should be padded with additional bytes up to multiple of 8 bytes. You might need to check about how the fulfillment should be done.

Encrypt the data

To encrypt the data,

- Ensure the key has been entered in the Key filed.
- Enter the plaintext data in the left (navy color) box.
- Then click the Encipher Data → button. The plaintext data will be calculated/encrypted and shown in the right (green color) box.

Decrypt the data

To decrypt the data,

- Ensure the key has been entered in the Key filed.
- Enter the to-be-decrypted data in the right (green color) box.
- Then click the ← Decipher Data button. The ciphered data will be calculated/decrypted and shown in the left (navy color) box.

Additional Information

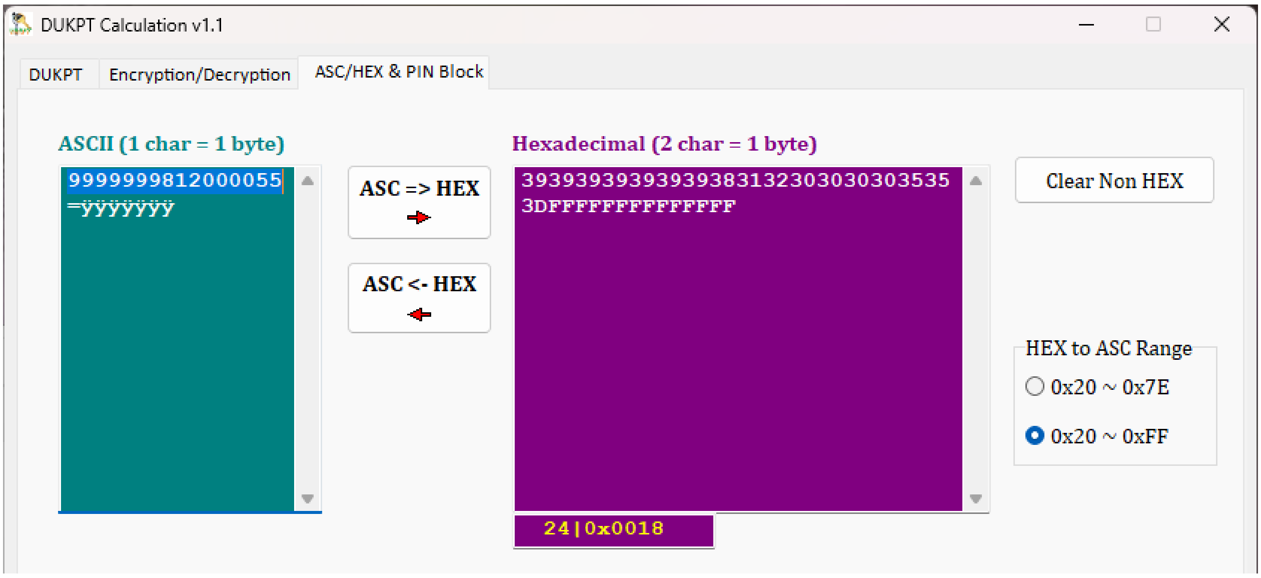
There is an information box located at the lower position of this tab. This is a description about how the MAC value should be calculated in Interac transactions against Elavon CA.

An MAC (**Massage Authentication Code**) must be generated for all Interac CT and CL transactions and sent to the Elavon CA host for validation. To generate the MAC correctly, the correct data and format must be prepared so the correct MAC can be generated. Of course, you also have to use the DUKPT MAC key for the MAC generation. The information box tells what kinds of data elements you should prepare for calculating the MAC.

5.3 Tab “ASC/HEX & PIN Block”

There are two utilities designed in this tab (see Figure 10):

(a) ASCII and Hexadecimal values conversion – upper portion
(b) PIN Block calculation - lower portion



ASCII (1 char = 1 byte)

```
9999999812000055
=ÿÿÿÿÿÿÿ
```

Hexadecimal (2 char = 1 byte)

```
39393939393939383132303030303535
3DFFFFFFFFFFFFFF
```

24|0x0018

HEX to ASC Range
○ 0x20 ~ 0x7E
● 0x20 ~ 0xFF

Figure 10. ASC/HEX format conversion (upper portion)

5.3.1 ASCII / Hexadecimal Data Type Conversion

In the payment transactions, the data may be encrypted in ASCII (1 character = 1 byte) or in Hexadecimal (2 characters = 1 byte) format. This utility is used for the format conversion between ASCII and Hexadecimal data.

- To convert from ASCII to Hexadecimal format, enter the data in the left (teal color) box and press the ASC => Hex → button. The result will be shown in the right (purple color) box.
- To convert from Hexadecimal to ASCII format, enter the data in the right (purple color) box and then press the ASC <- HEX ← button, the result will be shown in the left (teal color) box.

In the ASCII table, the characters between 0x20 ~ 0x7E are defined as printable characters and therefore they can be recognized easily. For the characters that are out of this range may not be shown correctly. So, you can select the HEX to ASC Range option to determine how to display the character format. For the characters that are out of the selected rang, each character will be replaced with the “dot (.)” character.

5.3.2 PIN Block calculation

5.3.2.1 Online PIN format

For security concerns, the Online PIN must be encrypted before it is sent out for verification. Before to send out the PIN, the PIN must be formatted into a **PIN block** before it is encrypted and sent out. The procedure to format a PIN block (refer to **Error! Reference source not found.**) is:

(1) Format the PIN data

- Fix the high nibble of Byte 1 (the leftmost byte) = 0 (or ‘0000’ in bits), to indicate the Format 0 rule is applied.
- Fill up the low nibble of Byte 1 with the length of the PIN digits
    - *Note: the PIN length must be 4 ≤ [PIN] ≤ 12*)
- Continue to fill in the PIN digits from byte 2 with BCD in the nibbles – high nibble first then low nibble. If the PIN length is odd, the lower nibble shall be filled with ‘F’
- For the unused nibbles, fill up with BCD ‘F’ to the end of Byte 8 (the rightmost byte)
- [Example] For PIN = ‘8482’, the PIN data shall be formatted as

  ’04 84 82 FF FF FF FF FF’

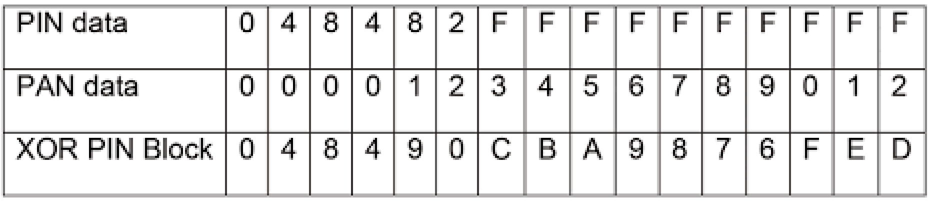
(2) Format the PAN data

- Forget the check digit of the PAN at the rightmost
- Pick the rightmost 12 digits and fill them between Byte3 to Byte 8, with high nibble first and low nibble next
- If the total length of the PAN is less than 12 digits (after the check digit was removed), keep the PAN to the right and fill up the pre-fix nibble with ‘0’.
- [Example] For PAN = “5671 2345 6789 0126”, the formatted PAN data =

  ‘00 00 12 34 56 78 90 12’

(3) Exclusive-OR the formatted PIN and PAN data at Step (1) and (2) to become the complete PIN block, which will be

  ’04 84 90 CB A9 87 6F ED’

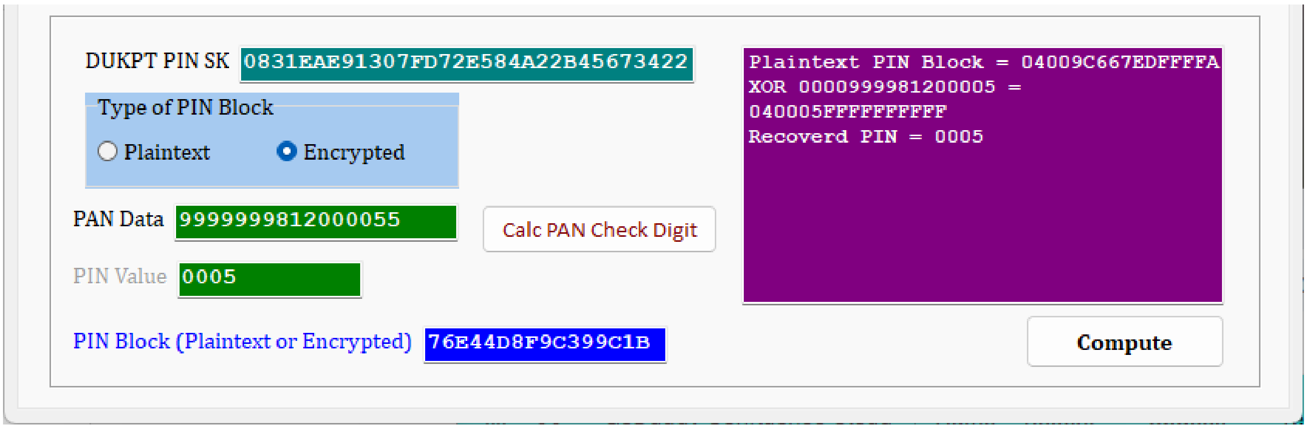


PIN data	0	4	8	4	8	2	F	F	F	F	F	F	F	F	F	F
PAN data	0	0	0	0	1	2	3	4	5	6	7	8	9	0	1	2
XOR PIN Block	0	4	8	4	9	0	C	B	A	9	8	7	6	F	E	D

Table 2 - Format of the PIN block before it is encrypted

5.3.2.2 PIN Block calculations

This utility can perform the PIN block encryption or PIN data recovery (see).



DUKPT PIN SK 0831EAE91307FD72E584A22B45673422

Type of PIN Block

○ Plaintext ● Encrypted

PAN Data 9999999812000055

PIN Value 0005

PIN Block (Plaintext or Encrypted) 76E44D8F9C399C1B

```
Plaintext PIN Block = 04009C667EDFFFFA
XOR 0000999981200005 =
040005FFFFFFFFFF
Recoverd PIN = 0005
```

Figure 11. PIN block encryption/decryption (lower portion)

PIN Block Encryption

(1) Select the **Plaintext** in the Type of PIN Block radiobutton group. The PIN Value filed will be enabled for this activity.

(2) Fill in the required data in the related fields:
   - DUKPT PIN SK – this should be derived in Sec 5.1. Generally, it will be filled in automatically.
   - PAN Data – you might have to decrypt the PAN (Tag '5A') or Track2 (Tag '57') data to recover the plaintext data
   - PIN Value

(3) Press the Compute button to get the encrypted PIN block. The calculation process will be shown in the purple box on the right, and the result will be shown in the PIN Block (Plaintext or Encrypted) field.

PIN Value Recovery

(1) Select the **Encrypted** in the Type of PIN Block radiobutton group.

(2) Fill in the required data in the related fields:
   - DUKPT PIN SK – this should be derived in Sec 5.1. Generally, it will be filled in automatically.
   - PAN Data – you might have to decrypt the PAN (Tag '5A') or Track2 (Tag '57') to retrieve the plaintext data.
   - PIN Block (Plaintext or Encrypted) – this data can be found in Tag '99'.

(3) Press the Compute button to recover the plaintext PIN value. The calculation process will be shown in the purple box on the right, as well as in the PIN Value field.

6. Examples

6.1 For sensitive data

	Data elements	Tag	Example data
1	BDK for data	n/a	49353F96231C16CB 4683E8F1E6EC69AF
2	KSN for data	1F8102	FFFF1122330041400F9
3	Derived DUKPT data key	n/a	81E013D074D9BADE 2444E7023D357203
4	Enciphered PAN	5A	142CE9E9587DF89B
5	Plaintext PAN	n/a	Decrypt Data [4] with the key [3] in CBC mode 4761730000000011
6	Enciphered Track 2	57	959C9B827CE8572B 95413CE440161948 9B1D16FB51CA914C 3B71C35212E1AC8C
7	Plaintext Track 2	n/a	Decrypt Data [6] with the key [3] in CBC mode 3437363137333030 3030303030303131 3D33313132323031 3133303331303036

6.2 For PIN block

	Data elements	Tag	Example data
1	BDK for PIN	n/a	7D99FCDE962E78E7 D2EE48F5ED3A848D
2	KSN for PIN	1F8101	FFFF654321006B20000B
3	Derived DUKPT PIN key	n/a	CD6A2ED6589935A4 45B2ACC4347B5DFA
4	Encrypted PIN block	99	CA09435FCE846242
5	Plaintext PIN block	n/a	Decipher Data [4] with the key [3] in CBC mode 06111859FFFFFFFD
6	Plaintext PAN	n/a	6210948000000029
7	Formatted PAN data	n/a	Formatted from Data [6] 0000094800000002
8	Recovered PIN data	n/a	(a) XOR Data [5] and [6], (b) remove the first byte and the padding characters 111111

6.3 For Interac MAC

	Data elements	Tag	Example data
1	BDK for Interac MAC	n/a	001E3F1E110C59B1 30C6C3B17E5C1DB1
2	KSN for Interac MAC	1F820B	FFFF9988910060E00012
3	Derived DUKPT MAC key	n/a	C777D71865B211EA 0BC27049E120B71F
4	Data elements for MAC		
	• Processing Code (CT)	1F8169	002000
	• Amount, Authorized	9F02	000000001300
	• STAN	1F8168	285068
	• PAN (plaintext)	5A	0012020000001
5	Catenated all DEs in Step [4] and converted to HEX format	n/a	3030323030302030 3030303030303031 3330302032383530 3638203030313230 3230303030303031
6	MAC calculation *(The MAC value is picked from the first 4 bytes of the last data block)*	n/a	Encrypt Data [5] with Key [3] in CBC mode (DES first, TDES at last) EA646006260EEF91 70BAC94B7A970426 65B69255C4B7A920 9818337F17F72068 84EC74D12CED4F8A

End of Document