E2E Encryption – 3DES DUKPT

Public - Security Level 0

March 2022

REVISION HISTORY

Date	Version	Author(s)	Comments
7/18/18	1.0	M. Billips	Initial version

CONFIDENTIALITY STATEMENT

Contents

E2E Encryption Overview

This guide contains specific information on how to pass encrypted track data to the EPX platform. Refer to the *EMV Reference Guide* and respective EPX Transaction Specifications for additional information on transaction types and request POST formatting.

End to End Encryption Identification

The E2EE tag (End to End Encryption) identifies which type of encryption is in use. Use this tag in correspondence with the CARD_ENT_METH tag.

E2EE value	Description
0	Use CARD_ENT_METH tag to identify Format
2	3DES Format (generic)

- Variable Type: Numeric
- Max Length: 1

Example:

```
<E2EE>2</E2EE>
```

Card Entry Method

The CARD_ENT_METH field is used to indicate how the card data was collected for the transaction and must be included in the request with the correct EPX value. The below table indicates which EPX value needs to be sent in correspondence to the device or application output.

Card Entry Method Code	Description
D	Used when sending Swiped Track 2 data
G	Chip (EMV contact)
H	Used when sending Swiped Track 1 data
Q	Proximity / Contactless MSD non-EMV (Track 1 or Track 2 allowed)
R	Proximity / Contactless EMV-based (Track 2 data is required)
X	Key Entered MICR or Card Number (Typed in using keyboard or keypad)

Device ID / Track Data Format Type (Component 1)

The Device ID / Track Data Format Type is the first component of the TRACK_DATA and will indicate the encryption mode / format type. The below table indicates the value for supported Encryption Types.

Encryption Type Value	Description
004	Generic 3DES - HEX Data 3DES DUKPT CBC Data Variant
005	Generic 3DES - BCD Data 3DES DUKPT CBC Data Variant

Track Data

The TRACK_DATA field in the EPX API must be present and requires the three components outlined below.

These components are concatenated together and placed in the EPX TRACK_DATA field.

Component 1

Contains the device ID / format type and requires a value of “004” or “005” for GENERIC 3DES DUKPT.

- Format: Numeric, zero filled, right justified.
- Position: 1 - 3
- Length: 3 bytes

Example:

```
004
```

Component 2

Contains the KSN (Key Serial Number). This value will be present in the data produced from the device output.

- Format: Hex
- Position: 4 - 23
- Length: 20 bytes

Example:

```
FFFF9876543210E0000D
```

Component 3

Contains the encrypted track data. This value will be present in the data produced from the device output.

- Format: Hex or BCD
- Position: 24 - N
- Length: Variable multiple of 16

Track Data Example (Generic 3DES - HEX Data 3DES DUKPT CBC Data Variant):

```
ED4A250ABE6343390B4D1036B4D550EBC9EB81CA4BAC012E7E7D2E654B6B83413D05A2F8C7E
EB902
```

Full String Example:

```
004FFFF9876543210E0000DED4A250ABE6343390B4D1036B4D550EBC9EB81CA4BAC012E7E7D
2E654B6B83413D05A2F8C7EEB902
```

Request POST Example:

The examples below contain Generic 3DES - HEX Data 3DES DUKPT CBC Data Variant Track Data. The **bold** text denotes mandatory fields to be sent in request.

Key Value Pair

```
E2EE=2
&CARD_ENT_METH=D
&TRACK_DATA=004FFFF9876543210E0000DED4A250ABE6343390B4D1036B4D550EBC9EB81CA
4BAC012E7E7D2E654B6B83413D05A2F8C7EEB902
&CUST_NBR=1234
&MERCH_NBR=1234567
&DBA_NBR=1
&TERMINAL_NBR=1
&TRAN_TYPE=CCR1
&AMOUNT=1.00
&BATCH_ID=1
&TRAN_NBR=6
&FIRST_NAME=EPX
&LAST_NAME=Test
&MAC=MAC9876543210
&CURRENCY_CODE=840
```

XML

```
<DETAIL CUST_NBR='1234' MERCH_NBR='1234567' DBA_NBR='1' TERMINAL_NBR='1'>
<AMOUNT>10.00</AMOUNT>
<BATCH_ID>1</BATCH_ID>
<TRAN_NBR>6</TRAN_NBR>
<TRAN_TYPE>CCR1</TRAN_TYPE>
<TRACK_DATA>004FFFF9876543210E0000DED4A250ABE6343390B4D1036B4D550EBC9EB81CA
4BAC012E7E7D2E654B6B83413D05A2F8C7EEB902</TRACK_DATA>
<CARD_ENT_METH>D</CARD_ENT_METH>
<E2EE>2</E2EE>
<CURRENCY_CODE>840</CURRENCY_CODE>
<MAC>MAC9876543210</MAC>
</DETAIL>
```

Ingenico RBA - Device Output

The data must be extracted from the Track 3 output from the Ingenico device and formatted correctly when sent to EPX in the request POST. The following outlines the specific detail structure of the output.

The Track 3 data sent to the POS consists of four items separated by colons (:):

1. The KSN of the TDES DUKPT encryption key: 20 bytes ASCII hex characters.
2. One digit indicating which data were encrypted: 1 = Track 1, 2 = Track 2, 3 = dummy tracks for manually-entered data, 4 = Track 1 and Track 2.
3. The four-digit length (decimal) of the encrypted data block. This is the number of bytes of binary data.
4. The encrypted data block in ASCII Hex format. Since each byte is represented by two ASCII characters, the length of this string will be twice the length of the binary data block.

The following is an example of Track 3 output containing Track 1 and Track 2 data for TDES DUKPT:

```
FFFF9876543210E00001:4:0088:45E7C9C1D89C32BA6342F2E405957A12FE053E00B4A0B43
6FC0629F10CFC0A55F724D4E35FB7C1D2E3C778164514E216DB58C4C112EA42E113CB31C152
433827A936A3A69BED84C001CF75769D6737FCC9E4A1620FC629A2
```

Ingenico-Specific Card Entry Method

The CARD_ENT_METH field in the EPX API must be present and mapped to the correct EPX value. The below table indicates which EPX value needs to be sent in correspondence to the Ingenico output value.

Ingenico value indicating what data was encrypted	**EPX CARD_ENT_METH**
1 = Track 1	H
2 = Track 2	D
3 = Dummy tracks for key entry	X
4 = Track 1 and Track 2	H