7 UPLOAD DUKPT FUNCTIONALITY

7.1 Introduction

DUKPT is an abbreviation of Derived Unique Key Per Transaction, a very descriptive name as will be explained in this chapter:

- A short introduction of DUKPT is given in section 7.2;
- The three DUKPT Use Cases supported by UpLoad are described in section 7.3;
- The UpLoad function Import TIK File will be described in section 7.4;
- The UpLoad function Generate TIK File will be described in section 7.5.

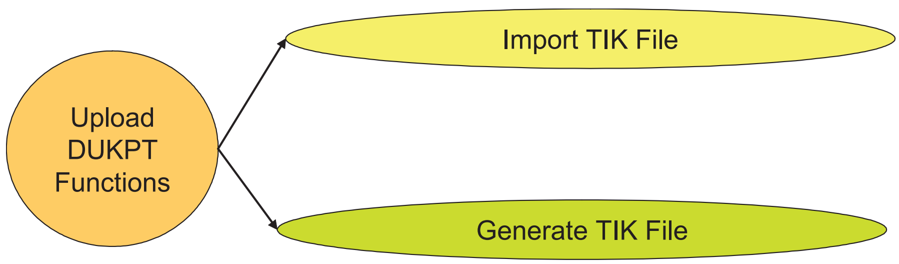

Figure 7.1 - UpLoad DUKPT Functions

7.2 Short Introduction of DUKPT

DUKPT is a security scheme used to secure the communication between a host system and a population of Target Devices. In the DUKPT security scheme, the host owns a single secret, the so-called Derivation Key (DK). This is a double length DES key. The DK will be used to calculate or derive a secret Terminal Initial Key (TIK) for the individual Target Devices. The TIK is also a double length DES key.

In football, each team has a specific shirt with logo and colours. The shirt is used to identify a team. Each player of a team has a unique number on the back of the shirt. This shirt number identifies a player within a team. This same principle is also used in the DUKPT security scheme:

- A Key Set Identifier (KSID) is used to identify a group of Target Devices. A KSID consists of 5 bytes or 40 bits, for example `12AB34CD56`. A group of Target Devices is the ‘team’. The KSID is the ‘shirt’ of the ‘team’;

- For each KSID or ‘team’ of Target Devices, there will be a double length DK. This DK will be used to generate a unique secret TIK for each individual Target Device that is part of the ‘team’;
- Each TIK has a unique number on the ‘shirt’. This number is called the Key Serial Number (KSN). The KSN consists of 19 bits;
- Each ‘action’ of a specific ‘player’ (KSN) of a specific ‘team’ (KSID) is also assigned a unique number: the so-called Transaction Counter (TC). This TC consists of 21 bits. In a Target Device an ‘action’ corresponds to a specific transaction. Typically, at the beginning of a new transaction, the TC will be incremented;

The combination of KSID, KSN and TC is called the Security Module Identifier (SMID) and consists of 40 + 19 + 21 = 80 bits = 10 bytes. See Figure 7.2:

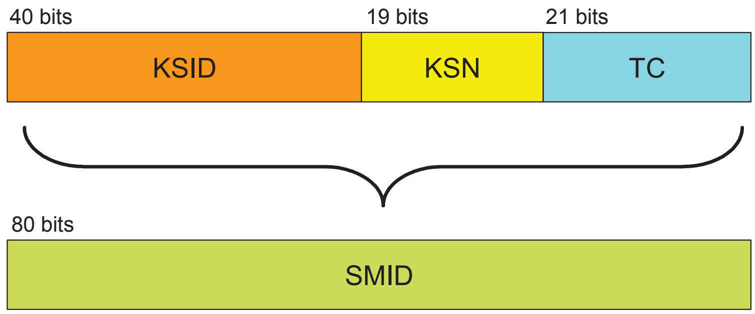

Figure 7.2 – SMID Structure

The SMID is exchanged between Target Device and host system during each transaction. Before a Target Device with support for DUKPT can be used in the field to secure EFT transactions, the Target Device must be loaded with:

- A unique SMID value;
- The corresponding double length TIK.

~~7.3 DUKPT Use Cases~~

~~7.3.1 Introduction~~

~~For the DUKPT security scheme, the following activities can be distinguished:~~

- ~~The generation of (SMID, TIK) pairs using a Derivation Key. Usually, this is done by the organisation that is responsible for the host system;~~
- ~~The loading of (SMID, TIK) pairs into the Target Devices. Usually this is done by the Supplier of the Target Devices.~~

~~There are three different DUKPT Use Cases that can be distinguished:~~

- ~~The DUKPT Import Use Case is described in paragraph 7.3.2 ;~~
- ~~The DUKPT Export Use Case is described in paragraph 7.3.3;~~
- ~~The DUKPT Real-Time Use Case is described in paragraph 7.3.4.~~

~~7.3.2 DUKPT Import Use Case~~

~~When the host organisation generates the (SMID, TIK) pairs and the Supplier loads a (SMID, TIK) pair in each Target Device, there is a need for a secure method to transfer (SMID, TIK) pairs from the host organisation to the Supplier. This can be done as follows:~~

- ~~The host organisation and Supplier share a so-called DUKPT Transport Key (TK);~~
- ~~A batch of (SMID, TIK) pairs will be generated by the host organisation and the generated batch of (SMID, TIK) pairs will be stored in a so-called TIK file;~~
- ~~The secret parts of the TIK file will be encrypted with the DUKPT Transport Key by the host organisation;~~
- ~~The TIK file containing a batch of (SMID, TIK) pairs will be send to the Supplier by the host organisation;~~
- ~~During DUKPT key loading at the Supplier premises, the DUKPT Transport Key shared with the host organisation can be used to decrypt the secret parts of the TIK file.~~

~~Name~~	~~Type~~	~~Value~~
		~~3FA85B7DE14DA02EB8B08E896DBFAA67~~
~~P1~~	~~008b~~	~~7DDA760C8610ECD0~~
~~P2~~	~~008b~~	~~CC9F25B158FA6ECE~~
~~(P3 XOR P4)~~	~~008b~~	~~8718D5F48CF20A49~~
~~E[MMTK] (P1)~~	~~008b~~	~~3B4DFF44C4E12723~~
~~E[MMTK] (P2)~~	~~008b~~	~~B3CB77BBEC0B79B7~~
~~E[MMTK] (P3 XOR P4)~~	~~008b~~	~~05B863E8C55282F9~~
~~MTK$_I$~~	~~024b~~	~~3B4DFF44C4E12723B3CB77BBEC0B79B7~~ ~~05B863E8C55282F9~~
~~KVC of MTK$_I$~~	~~002b~~	~~71A3~~

~~16.5.2 Derivation of the K-SC$_i$ and K-OPP-B$_i$~~

~~The K-SC$_i$ is derived from the K-SC using the following algorithm:~~

- ~~Take the 11 characters of the Target Device Serial Number in ASCII representation, for example ‘209-133-065’;~~
- ~~Calculate a SHA-1 over the 11 characters of the Target Device Serial Number. The result of this operation consists of 20 bytes;~~
- ~~P1 is the leftmost 8 bytes of the SHA-1 Result;~~
- ~~P2 is the rightmost 8 bytes of the SHA-1 Result;~~
- ~~Encrypt P1 with K-SC. The result is the left part of the derived key K-SC$_i$;~~
- ~~Encrypt P2 with K-SC. The result is the right part of the derived key K-SC$_i$.~~

~~The following table contains an example of the K-SC$_i$ derivation:~~

~~Name~~	~~Type~~	~~Value~~
~~Derivation Algorithm~~	~~002n~~	~~02 = K-SC$_I$ derivation~~
~~K-SC~~	~~016b~~	~~01326754CDFEAB98CFDFEFFF8F9FAFBF~~
~~KVC of K-SC~~	~~002b~~	~~3B7C~~
~~Target Device S/N~~	~~011a~~	~~209-133-065~~
~~SHA-1 Result~~	~~020b~~	~~2E14A450E11AF5D7A20C236DED25B2B280597596~~
~~P1~~	~~008b~~	~~2E14A450E11AF5D7~~
~~P2~~	~~008b~~	~~ED25B2B280597596~~
~~E[K-SC] (P1)~~	~~008b~~	~~4A81C0C60BE3DB89~~
~~E[K-SC] (P2)~~	~~008b~~	~~0CB2E9FCC5BD32AA~~
~~K-SC$_I$~~	~~016b~~	~~4A81C0C60BE3DB890CB2E9FCC5BD32AA~~
~~KVC of K-SC$_I$~~	~~002b~~	~~3203~~

16.5.3 Derivation of DUKPT TIK

The calculation of the TIK is done in a number of steps:

- Calculate an input block of 8 bytes = 64 bits from the following fields:
  - KSID = 5 bytes is 40 bits;
  - DUKPT KSN = 19 bits;
  - Filler of 5 bits with value 00000_b.

- The left part of the DUKPT TIK is the result of encryption of this input block with the DUKPT Derivation Key;
- Calculate a variant of the DUKPT DK by exclusive ORing the DUKPT DK with a constant with value `C0C0C0C000000000C0C0C0C000000000`;
- The right part of the DUKPT TIK is the result of encryption of this input block with the DUKPT Derivation Key variant;
- Concatenate left and right part of the DUKPT TIK.

An example:

Name	Type	Value
Key Derivation Algorithm	002n	05 = DUKPT TIK derivation
KSID	005b	CCCC020406
DK	016b	C1EFF87983FDE3D9B3237F852C1C43B3
KVC of DK	002b	00BA
DUKPT KSN	001n	1 = 000000000000000001_b
Input for encryption	008b	CCCC020406000020
Left part of TIK	008b	9B8EB4A6747EA849
Constant	016b	C0C0C0C000000000C0C0C0C000000000
DK variant	016b	012F38B983FDE3D973E3BF452C1C43B3
KVC of DK variant	002b	73A1
Right part of TIK	008b	AB1941D9A7289B38
TIK	016b	9B8EB4A6747EA849AB1941D9A7289B38
KVC of TIK	002b	E571

The following table contains another example:

Name	Type	Value
Key Derivation Algorithm	002n	05 = DUKPT TIK derivation
KSID	005b	CCCC020406
DK	016b	C1EFF87983FDE3D9B3237F852C1C43B3
KVC of DK	002b	00BA
DUKPT KSN	001n	2 = 000000000000000010_b
Input for encryption	008b	CCCC020406000040
Left part of TIK	008b	FEA567BADA30CD55
Constant	016b	C0C0C0C000000000C0C0C0C000000000
DK variant	016b	012F38B983FDE3D973E3BF452C1C43B3
KVC of DK variant	002b	73A1
Right part of TIK	008b	6B5F17F3A0AF7C1F
TIK	016b	FEA567BADA30CD556B5F17F3A0AF7C1F
KVC of TIK	002b	5EDB

16.5.4 Derivation of Double Length TMK_i

The calculation of a Double Length TMK_i is done as follows:

- Take the Target Device Serial Number;