Understanding PIN Pad Encryption/Decryption

For K100 testing, you will find this demo app helpful: 80114806-002-A Sftw;SDK;SmartPIN K100 PCI 3.X.zip

**Question:** How can I verify that my K100 is sending encrypted PIN data? And: How can I decrypt a PIN?

**Answer:** You can use our standard tools (see above and below) to decrypt a PIN. Note that when a host (which can be a reader) sends a Get Encrypted PIN command to the PIN pad, the host has to send not only the command itself, but the card's PAN. You need to know the PAN in order to obtain, and recover (through decryption), the PIN. The steps needed to get the decrypted PIN are discussed below.

Note that ID TECH follows the industry convention of adhering to ISO-9564, Format 0, with regard to PIN block encryption.

Also note that the example given below will use a PAN value of 01234567890123456 and a PIN of 9876. The final PAN digit, 6, is disregarded (in accordance with ISO-9564, Format 0, which says to ignore the PAN's "check digit"). Hence, when we say, further below, to use the "final 12 digits" of the PAN, we mean to use the portion of PAN corresponding to 456789012345 – the final 12 digits, *not including* the checksum digit.

### Overview

When the card reader pauses to obtain the user's PIN, the reader must send the card number (PAN) to the PIN pad along with a KSN. The PIN pad uses the KSN to derive a one-time PIN key. The PAN is then combined with the PIN (using XOR), and the combination (called a "PIN block") is encrypted with the one-time key before being sent to the reader. The procedure outlined below tells how to decrypt the PIN block.

### Procedure

First be sure your K100 is in anti-tamper (Removal Detection) mode by following the instructions given in SmartPIN K100 Removal Detection Instructions 08162012.pdf.

Next, launch the demo app (from the zipfile above) with the K100 connected to your computer. Use the Open Port button (top left, demo app main window) to verify that the K100 is talking to the demo app.

To verify that the K100 is encrypting PIN data properly, you need to enter a PIN manually and obtain the encrypted version of it. Here is how you do it:

Do the following:

1. Click the Get PIN button in the demo.
2. Enter a PIN (4 to 12 digits) on the K100 keypad, and then press the ENT (Enter) button. For this example, we are going to use a 4-digit PIN of 9876.
3. In the Result field of the demo app (bottom of window), you will see 36 hex nibbles. For example: 6299490079000060000F73D95376351257C8 (see screen shot).
The first 20 letters are the KSN. E.g., the boldface letters below represent the KSN in this example.

6299490079000060000F
73D95376351257C8

The encrypted PIN data can be found in the trailing 16 nibbles: 73D95376351257C8

How to Decrypt a PIN

You can use the EncryptionDecryptionTool.html to decrypt the PIN, assuming your K100 was injected with a key whose BDK value you know (such as the standard test key: 0123456789ABCDEFFEDCBA9876543210).

First, derive the decryption key by using the Derive button (in the Encryption/Decryption Tool) to derive a PIN Key Variant DUKPT key using the KSN obtained as described above. For example:
This combination of BDK and KSN will give a PIN-variant DUKPT key of A8CDB1D5FE9C2ED95D1C06D0451DB769 when you click the Derive Key button.

In the above example, our data trailer (final 16 nibbles) is 73D95376351257C8. Paste this value into the Data field of the Encryption/Decryption Tool.

Use the Decrypt button to TDES-decrypt the data trailer, using the PIN-variant key you derived. In this example, the decrypted data looks like: 0498339876FEDCBA.

Notice that the first two nibbles (04) specify the length of the PIN. In this case, the length is 4.

The next two nibbles (98) are the first two digits of the PIN itself.

To get the rest of the PIN, you need to XOR the remainder of the decrypted data trailer (339876FEDCBA) with the final 12 digits of the PAN, not including the check digit. Since the test PAN is 01234567890123456, this means you will XOR 456789012345 (hex) with 339876FEDCBA (hex). If you are using the Encrypt/Decrypt Tool in Chrome, go to that tool and open a console with Control-Shift-J, then cut and paste the following code into the console and hit Enter.

```javascript
PAN = "456789012345";
data = "339876FEDCBA";
custom.log(XORdataHex(PAN, data));
```

(This code only works if the Encrypt/Decrypt Tool is open and has focus. The JavaScript logic relies on code in the tool.) This code will do the XOR operation for you. The result will be (in this case) 76FFFFFFFE. (You can get this value, also, by using a "Programmer’s Calculator" to XOR the nibbles one by one: XOR 4 with 3, then 5 with 3, then 6 with 9, etc.)

The repeating F values are padding. What remains after removing padding is 76. Combine this with the 98 obtained before to obtain the final PIN: 9876.

For more about PIN encryption, see https://en.wikipedia.org/wiki/ISO_9564#PIN_encryption. This describes ISO-9564, the relevant industry standard.

NOTE: In this ID TECH document, Jane Xu explains that when a PAN is less than 16 digits long, it gets zero-padded on the left for purposes of PIN recovery. When the PAN is more than 16 digits, only the rightmost 16 digits are used.