USER MANUAL

iMag

Magnetic Stripe Reader for iPhone 3G, 3GS and iPod Touch

80097503-001
06/10/2010

1 iMag Firmware Command

1.1 *Setting Command*

The setting data command is a collection of many function setting blocks and its format is as follows.
Command:
<STX><S><FuncSETBLOCK1>…<FuncBLOCKn><ETX><LRC>
Response: <ACK> or <NAK> for wrong command (invalid funcID, length and value)

Each function-setting block <FuncSETBLOCK> has following format:

<FuncID><Len><FuncData>

Where:
<FuncID> is one byte identifying the setting(s) for the function.
<Len> is a one byte length count for the following function-setting block <FuncData>.
<FuncData> is the current setting for this function. It has the same format as in the sending command for this function.
Example: Set Dukpt key management
CMD: 02 53 58 01 31 03 3A
OUT: 06

1.2 *Get Firmware Version*

Get Firmware Version command will respond a firmware version to application.
Command: <STX><R><FmVerID><ETX><LRC 1>
Response: <ACK> <STX><Version String><ETX><LRC 2>

Version String will be in format of “ID TECH iMag Swipe Reader x.y.z” x.y.z is the major and minor version number.

1.3 *Get Setting*

This command will send current setting to application.
Command: <STX> <R> <ReviewID> <ETX> <LRC 1>
Response: <ACK> <STX> <FuncID> <Len> <FuncData> <ETX> <LRC 2>

<FuncID>, <Len> and <FuncData> definition are same as described above.
Example: Review all setting

CMD: \02\52\1F\03\4C
OUT: \06\02\7E\01\31\4C\01\31\58\01\31\03\5B

1.4 *Function ID Table*

The default setting is shown in **bold**.

Function Name	Function ID	Description
EncryptionID	0x4C	Security Algorithm **'0' Clear Text** '1' Triple DES '2' AES
keyManagementID	0x58	**'0' DUKPT** '1' Fixed Key
SecurityLevelID	0x7E	Security Level (Read Only) '0' ~ '3" **Default value '1'**
GetFirmwareVersion	0x22	returns current firmware version

2 Data Output Format

2.1 *iMag Unencrypted Data Output Format*

Track 1: <Start Sentinel 1><T_1 Data><End Sentinel><Track Separator>
Track 2: <Start Sentinel 2><T_2 Data><End Sentinel><Track Separator>
Track 3: <Start Sentinel 3><T_3 Data><End Sentinel><Terminator>

where: Start Sentinel 1 = %
Start Sentinel 2 = ;
Start Sentinel 3 = ; for ISO, % for AAMVA
End Sentinel all tracks = ?

Start or End Sentinel: Characters in encoding format which come before the first data character (start) and after the last data character (end), indicating the beginning and end, respectively, of data.

Track Separator: A designated character which separates data tracks. The default character is CR (Carriage Return).

Terminator: A designated character which comes at the end of the last track of data, to separate card reads. The default character is CR (Carriage Return).

For example:

```
%B4352378366824999^TFSTEST /THIRTYONE
^0510201100008820088200000?;4352378366824999=051020110000882?
```

2.2 *iMag Encrypted Data Output Format*

For ISO card, both clear and encrypted data are sent. For other card, only clear data are sent.

A card swipe returns the following data:

Card data is sent out in format of
<STX><LenL><LenH><Card Data><CheckLRC><CheckSum><ETX>

<STX> = 02h, <ETX> = 03h
<LenL><LenH> is a two byte length of <Card Data>.
<CheckLRC> is a one byte Exclusive-OR sum calculated for all <Card Data>.
<CheckSum> is a one byte Sum value calculated for all <Card data>.
<Card Data> format is

ISO/ABA Data Output Format:

- card encoding type (0: ISO/ABA)
- track status (bit 0,1,2:T1,2,3 decode, bit 3,4,5:T1,2,3 sampling)
- track 1 unencrypted length (1 byte in binary, 0 for no track1 data)
- track 2 unencrypted length (1 byte in binary, 0 for no track2 data)
- track 3 unencrypted length (1 byte in binary, 0 for no track3 data)
- track 1 masked
- track 2 masked
- track 3 data
- track 1 and track 2 encrypted (AES/TDES encrypted data)
- track 1 hashed (20 bytes SHA-1)
- track 2 hashed (20 bytes SHA-1)
- DUKPT serial number (10 bytes)

Non ISO/ABA Data Output Format:

- card encoding type (1: AAMVA, 2: CADL, 3: Others)
- track status (bit 0,1,2:T1,2,3 decode, bit 3,4,5:T1,2,3 sampling)
- track 1 length (1 byte in binary, 0 for no track1 data)
- track 2 length (1 byte in binary, 0 for no track2 data)
- track 3 length (1 byte in binary, 0 for no track3 data)
- track 1 data
- track 2 data
- track 3 data

Description:

Track 1 and Track 2 unencrypted Length

This one-byte value is the length of the original Track data. It indicates the number of bytes in the Track masked data field. It should be used to separate Track 1 and Track 2 data after decrypting Track encrypted data field.

Track 3 unencrypted Length

This one-byte value indicates the number of bytes in Track 3 masked data field.

Track 1 and Track 2 masked

Track data masked with the MaskCharID (default is '*'). The first PrePANID (up to 6 for BIN, default is 4) and last PostPANID (up to 4, default is 4) characters can be in the clear (unencrypted).

Track 1 and Track 2 encrypted

This field is the encrypted Track data, using either TDES-CBC or AES-CBC with initial vector of 0. If the original data is not a multiple of 8 bytes for TDES or a multiple of 16 bytes for AES, the reader right pads the data with 0.
The key management scheme is DUKPT. The key for encrypting data is Pin Encryption Key (a DUKPT term). It is the DUKPT Derived Key Xor with 00000000000000FF00000000000000FF.

How to get Encrypted Data Length

Track 1 and Track 2 data are encrypted as a single block. In order to get the number of bytes for encrypted data field, we need to get Track 1 and Track 2 unencrypted length first. The field length is always a multiple of 8 bytes for TDES or multiple of 16 bytes for AES. This value will be zero if there was no data on both tracks or if there was an error decoding both tracks. Once the encrypted data is decrypted, all padding 0 need to be removed. The number of bytes of decoded track 1 data is indicated by track 1 unencrypted length field. The remaining bytes are track 2 data, the length of which is indicated by track 2 unencrypted length filed.

Track 1 hashed and Track 2 hashed

iMag reader uses SHA-1 to generate hashed data for both track 1 and track 2 unencrypted data. It is 20 bytes long for each track. This is provided with two purposes in mind: One is for the host to ensure data integrity by comparing this field with a SHA-1 hash of the decrypted Track data, prevent unexpected noise in data transmission. The other purpose is to enable the host to store a token of card data for future use without keeping the sensitive card holder data. This token may be used for comparison with the stored hash data to determine if they are from the same card.

Example:

1) The following is the data sent out from the reader when swiping a ISO/ABA Data Output Format (eg. credit card) and the reader is in the encrypted mode (security level=3):

```
02 05 01 00 1B 43 23 00 25 2A 34 33 35 32 2A 2A   .....C#.%*4352**
2A 2A 2A 2A 2A 2A 34 39 39 39 5E 54 46 53 54 45   ******4999^TFSTE
53 54 20 2F 54 48 49 52 54 59 4F 4E 45 20 5E 2A   ST /THIRTYONE ^*
2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A   ****************
2A 2A 2A 2A 2A 2A 2A 2A 2A 3F 2A 3B 34 33 35 32   *********?*;4352
2A 2A 2A 2A 2A 2A 2A 2A 34 39 39 39 3D 2A 2A 2A   ********4999=***
2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 3F 2A C3 25   ************?*Ã%
F6 26 F1 65 D9 CC F2 E1 D1 74 BB 8A 80 1B D3 BC   ö&ñeÙÌòáÑt»Š .Ó¼
AC ED 19 1F 8A 24 CA 24 2B AD 54 3D 84 4B C7 DC   ¬í..Š$Ê$+T=„KÇÜ
3F 1F 3D 9C 9C 49 39 9C F5 98 E1 32 C7 57 3B B6   ?.=ææI9œõ˜á2ÇW;¶
A8 99 FB CE A9 45 A7 D8 98 26 D4 E5 EC 1C FC CD   ¨™ûÎ©E§Ø˜&Ôåì.üÍ
D5 EA 6C 92 C9 EC F8 D3 15 4F E8 A8 90 4A 9E FF   Õêl'ÉìøÓ.Oè¨ J ÿ
6B CB A6 49 B8 0D C4 0D 6C 59 07 F8 4A 24 E9 33   kË¦I¸.Ä.lY.øJ$é3
```

```
28 ED C8 C8 BE D4 02 12 DF DA 42 D8 7A 83 57 D1   (íÈȾÔ..ßÚBØzƒWÑ
9F 13 4E 07 5A BF 4A 8F 4D 1F D3 42 01 C2 2F 83   Ÿ.N.Z¿J.M.ÓB.Â/ƒ
55 1B 54 CB 2C D0 FB 90 1E 98 B9 37 D5 06 00 00   U.TË,Ðû.˜¹7Õ...
00 00 00 00 00 00 00 00 ED 35 03                  ........í5.
```

In this example,
LenL: 0x05
LenH: 0x01
Card encoding type: 0x00
Track status: 0x1B
Track 1 unencrypted length: 0x43
Track 2 unencrypted length: 0x23
Track 3 unencrypted length: 0x00
Track 1 masked:

```
25 2A 34 33 35 32 2A 2A 2A 2A 2A 2A 2A 2A 34 39 39 39 5E 54 46 53 54 45
53 54 20 2F 54 48 49 52 54 59 4F 4E 45 20 5E 2A 2A 2A 2A 2A 2A 2A 2A 2A
2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 3F 2A
```

Track 2 masked:

```
3B 34 33 35 32 2A 2A 2A 2A 2A 2A 2A 2A 34 39 39 39 3D 2A 2A 2A 2A 2A 2A
2A 2A 2A 2A 2A 2A 2A 2A 3F 2A
```

Track 3 data: nil

Track 1 and track 2 encrypted:

```
C3 25 F6 26 F1 65 D9 CC F2 E1 D1 74 BB 8A 80 1B D3 BC AC ED 19 1F 8A 24
CA 24 2B AD 54 3D 84 4B C7 DC 3F 1F 3D 9C 9C 49 39 9C F5 98 E1 32 C7 57
3B B6 A8 99 FB CE A9 45 A7 D8 98 26 D4 E5 EC 1C FC CD D5 EA 6C 92 C9 EC
F8 D3 15 4F E8 A8 90 4A 9E FF 6B CB A6 49 B8 0D C4 0D 6C 59 07 F8 4A 24
E9 33
28 ED C8 C8 BE D4
```

Track 1 hashed:

```
02 12 DF DA 42 D8 7A 83 57 D1 9F 13 4E 07 5A BF 4A 8F 4D 1F
```

Track 2 hashed:

```
D3 42 01 C2 2F 83 55 1B 54 CB 2C D0 FB 90 1E 98 B9 37 D5 06
```

DUKPT serial number:

```
00 00 00 00 00 00 00 00 00 00
```

2) The following is the data sent out from the reader when swiping a Non ISO/ABA Data Output Format and the reader is in the encrypted mode (security level=3):

```
02 E9 00 04 3F 51 28 6B 25 54 52 41 43 4B 31 37   .é..?Q(k%TRACK17
36 37 36 37 36 30 37 30 37 30 37 37 36 37 36 37   6767607070776767
36 30 37 30 37 30 37 37 36 37 36 37 36 30 37 30   6070707767676070
37 30 37 37 36 37 36 37 36 30 37 30 37 30 37 37   7077676760707077
36 37 36 37 36 30 37 30 37 30 37 37 36 37 36 37   6767607070776767
36 30 37 30 37 31 35 3F 47 3B 32 31 32 31 32 31   6070715?G;212121
32 31 32 31 37 36 37 36 37 36 30 37 30 37 30 37   2121767676070707
37 36 37 36 37 36 37 36 32 31 32 31 32 31 32 3F   767676762121212?
30 3B 33 33 33 33 33 33 33 33 33 33 37 36 37 36   0;33333333337676
```

```
37 36 30 37 30 37 30 37 37 36 37 36 37 36 33 33   7607070776767633
33 33 33 33 33 33 33 33 37 36 37 36 37 36 30 37   3333333376767607
30 37 30 37 37 36 37 36 37 36 33 33 33 33 33 33   0707767676333333
33 33 33 33 37 36 37 36 37 36 30 37 30 37 30 37   3333767676070707
37 36 37 36 37 36 33 33 33 33 33 33 33 33 33 33   7676763333333333
37 36 37 36 37 36 30 37 30 37 3F 32 53 F1 03 FF   767676070?2Sñ.ÿ
```