# SecureMag
# Encrypted MagStrip Reader
# User Manual

## USB, RS232 and PS2 Interface

80096504-001

4 October 2019

ID TECH
10721 Walker Street, Cypress, CA 90630-4720
Tel: (714) 761-6368  Fax (714) 761-8880
www.idtechproducts.com support@idtechproducts.com

## FCC WARNING STATEMENT

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his expense.

## FCC COMPLIANCE STATEMENT

This device complies with Part 15 of the FCC Rules. Operation of this device is subject to the following conditions: this device may not cause harmful interference and this device must accept any interference received, including interference that may cause undesired operation.

## CANADIAN DOC STATEMENT

This digital apparatus does not exceed the Class B limits for radio noise for digital apparatus set out in the **Radio Interference Regulations of the Canadian Department of Communications**.

Le présent appareil numérique n’émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de las classe A prescrites dans le Réglement sur le brouillage radioélectrique édicté par les ministère des Communications du Canada.

## CE STANDARDS

An independent laboratory performed testing for compliance to CE requirements. The unit under test was found compliant to Class B

## FCC warning statement

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

The user manual for an intentional or unintentional radiator shall caution the user that changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

**Note:** The grantee is not responsible for any changes or modifications not expressly approved by the party responsible for compliance. Such modifications could void the user's authority to operate the equipment.

**Note:** This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This device complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. The antenna(s) used for this transmitter must not be co-located or operating in conjunction with any other antenna or transmitter and must be installed to provide a separation distance of at least 20cm from all persons.

## Cautions and Warnings

| | |
|---|---|
| ⚠ | **Caution**: The ViVOpay Vendi should be mounted 1-2 feet away from other ViVOpay Vendi. Can be adjusted based on lane setup. |
| ⚠ | **Caution**: Danger of Explosion if battery is incorrectly replaced. Replace only with same or equivalent type recommended by the manufacturer. Discard used batteries according to the manufacturer's instructions. |
| ⚡ | **Warning**: Avoid close proximity to radio transmitters which may reduce the ability of the reader. |

## LIMITED WARRANTY

ID TECH warrants to the original purchaser for a period of 12 months from the date of invoice that this product is in good working order and free from defects in material and workmanship under normal use and service. ID TECH's obligation under this warranty is limited to, at its option, replacing, repairing, or giving credit for any product that returned to the factory of origin with the warranty period and with transportation charges and insurance prepaid, and which is, after examination, disclosed to ID TECH's satisfaction to be defective. The expense of removal and reinstallation of any item or items of equipment is not included in this warranty. No person, firm, or corporation is authorized to assume for ID TECH any other liabilities in connection with the sales of any product. In no event shall ID TECH be liable for any special, incidental or consequential damages to purchaser or any third party caused by any defective item of equipment, whether that defect is warranted against or not. Purchaser's sole and exclusive remedy for defective equipment, which does not conform to the requirements of sales, is to have such equipment replaced or repaired by ID TECH. For limited warranty service during the warranty period, please contact ID TECH to obtain a Return Material Authorization (RMA) number & instructions for returning the product.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES OF MERCHANTABILITY OR FITNESS FOR PARTICULAR PURPOSE. THERE ARE NO OTHER WARRANTIES OR GUARANTEES, EXPRESS OR IMPLIED, OTHER THAN THOSE HEREIN STATED. THIS PRODUCT IS SOLD AS IS. IN NO EVENT SHALL ID TECH BE LIABLE FOR CLAIMS BASED UPON BREACH OF EXPRESS OR IMPLIED WARRANTY OF NEGLIGENCE OF ANY OTHER DAMAGES WHETHER DIRECT, IMMEDIATE, FORESEEABLE, CONSEQUENTIAL OR SPECIAL OR FOR ANY EXPENSE INCURRED BY REASON OF THE USE OR MISUSE, SALE OR FABRICATIONS OF PRODUCTS WHICH DO NOT CONFORM TO THE TERMS AND CONDITIONS OF THE CONTRACT.

The information contained herein is provided to the user as a convenience. While every effort has been made to ensure accuracy, ID TECH is not responsible for damages that might occur because of errors or omissions, including any loss of profit or other commercial damage, nor for any infringements or patents or other rights of third parties that may result from its use. The specifications described herein were current at the time of publication but are subject to change at any time without prior notice.

ID TECH and Value through Innovation are trademarks of International Technologies & Systems Corporation. USB (Universal Serial Bus) specification is copyright by Compaq Computer Corporation, Intel Corporation, Microsoft Corporation, and NEC Corporation. Windows is registered trademarks of Microsoft Corporation.

ID TECH
10721 Walker Street
Cypress, CA 90630
(714) 761-6368

# Table of Contents

# 1. Introduction

ID TECH's SecureMag prevents card holder information from being accessed when data is in-transit or stored resulting in secure end to end transactions. The SecureMag reader delivers superior reading performance with its ability to encrypt sensitive card data. The reader fully supports TDES and AES data encryption using **DUKPT Key** management method. The SecureMag is offered in USB, RS232, and PS2 interfaces.

## 1.1. Document Notations

Notations used throughout the document:

- **Bold**: is a boldface font indicates default setting value.
- '2': is a single quotation indicates ASCII characters, for example, '2' is 32 in hex.
- "Number": is a null terminated character string.
- <Len>: are angle brackets that indicate a specific character or character string in a command or response.
- Hex: the hex character 53 is '5' in ASCII or 83 in decimal. Sometimes hex characters are represented with an h attached to the end, for example, 53h.
- \02: is a way to show that the following number is in hex. It is used by the configuration program.

# 2. Features and Benefits

- Bi-directional card reading
- Reads encoded data that meets ANSI, ISO, and AAMVA standards
- Custom formats such as ISO track 1 format on track 2 or 3
- Reads three tracks of card data
- An LED and beeper that provides the status of the reading operations.
- Compatible with USB specification Revision 2.0 (USB interface)
- Compatible with HID specification Version 1.1 (USB interface)
- Uses standard Windows HID driver for communications; no third-party device driver is required (USB interface)
- Provides clear text confirmation data including card holder's name and a portion of the PAN as part of the **Masked Track Data**
- User friendly software for device configuration.

# 3. Applicable Documents

| | |
|---|---|
| **ISO 7810 – 1985** | Identification Cards – Physical |
| **ISO 7811 - 1 through 6** | Identification Cards - Track 1 through 3 |
| **ISO 7816 - 1 through 4** | Identification Cards - Integrated circuit cards with contacts |
| **ISO 4909** | Magnetic stripe content for track 3 |
| **ISO 7812** | Identification Cards – Identification for issuers Part 1 & 2 |
| **ISO 7813** | Identification Cards – Financial Transaction Cards |
| **ANSI X.94** | Retail Financial Services Symmetric Key Management |

# 4. Specifications

| | |
|---|---|
| **Power Consumption** | 5VDC +/- 10%<br>Maximum operating consumption is less than 50mA<br>**RS232 interface**: external power adaptor that supplies power through RS232 cable.<br>**USB interface**: Is from host interface and no external power adaptor needed. |
| **Swipe Speed** | 3 to 65 inches per second<br>Bi-directional |
| **Indicators** | Tri-color LED<br>• The LED is off while reading and decoding.<br>• The red indicates bad read.<br>• The green indicates a good read, or the machine is ready.<br>Beeper<br>• A beeping sound indicates good read. |
| **LED Indicator** | 2mmx5mm, Green/Red dual color under firmware control |
| **Communication Interface** | RS232<br>• Baud Rate – 1200, 2400, 4800, 9600, 19200, 38400, 56700, 115200<br>• Data bits – 8<br>• Stop bits – 1 or 2<br>• Parity – off, odd, even, mark, or space<br>• Supports RTS/CTS hardware and Xon-Xoff software handshaking<br>USB<br>• Complies with USB 2.0 specification<br>PS2 Keyboard<br>• IBM PS2 interface compatible |
| **Card Size** | Supports cards that meet the ISO 7810 and 7811 1-7 standards. |
| **Dimension** | 3.94 inches (length) by 1.38 inches (width) and 1.18 inches (height). |

## 4.1. Interface Cable and Connector

RS232 interface:

- IDT standard **RS232 Interface Cable**
- DB-9 Female connector with 2mm power jack in the housing
- Standard cable length is 6 feet

**Pin Out Table**

| J1* | Color | Signal | P1* |
|---|---|---|---|
| **1** | - | CASE_GND | SHELL |
| **2** | White | TXD | 2 |
| **3** | Green | RXD | 3 |
| **4** | Yellow | VCC | From power jack |
| **5** | Brown | RTS | 8** |
| **6** | Grey | CTS | 4** |
| **7** | Black | GND | 5 |

***J1 is the connector to PCB end and P1 is DB-9 end**

**** RTS and CTS are not used unless hardware handshaking support is enabled by Function ID 0x44 (Handshake)**

USB

- IDT standard USB interface cable
- Series "A" plug
- Standard cable length is 6 feet

**Pin Out Table**

| J1 | Color | Signal | P1 |
|---|---|---|---|
| **1** | - | CASE_GND | SHELL |
| **3** | GRN | +DATA | 3 |
| **5** | Red | V_IN | 1 |
| **6** | White | -DATA | 2 |
| **7** | BLK | GND | 4 |

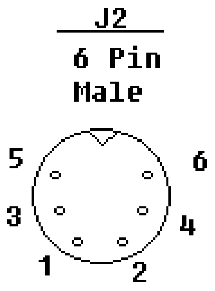
### 4.1.1. Keyboard Wedge

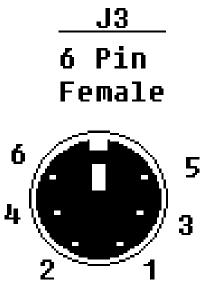
- IDT standard Keyboard Wedge cable.
- Y cable with dual PS/2 6-pin mini-DIN connectors; male side is connected to PC, female side connected to KB.
- Standard cable length is 6 feet.

**Pin Out Table**

| J1 | Color | Signal | J2 | J3 |
|---|---|---|---|---|
| **1** | - | CASE_GND | SHELL | SHELL |
| **2** | White | P-CLK | 5 | -- |
| **3** | Green | P-DATA | 1 | -- |
| **4** | Yellow | VCC | 4 | 4 |
| **5** | Brown | K-CLK | -- | 5 |
| **6** | Grey | K-DATA | -- | 1 |
| **7** | Black | GND | 3 | 3 |

### 4.1.2. PS/2 Connector





# 5. Operations

The magnetic stripe must be facing towards the magnetic read head and may be swiped in either direction. A card may be swiped through the reader slot when the LED is green. After swiped, the LED will blank until the decoding process is completed. If the transaction is accepted, the reader will return green. If declined, the reader will flash red before turning green again.

During a data transfer the LED will be powered off but will function again when the light returns to green. A red LED indicates an error. The beeper will provide error indications by beeping for each correctly read data track. Depending on the security level configured, the card data might be displayed in clear or encrypted mode.

# 6. Command Process

**Command Requests** and **Responses** are sent and received from the device. For USB interface devices, the commands are sent to the device using HID class specific request **Set_Report (`21 09` …)**. The response to a command is retrieved from the device using a HID class specific request **Get_Report (`A1 01` …)**. These requests are sent over the default control pipe.

For RS232 interface devices, please see the commands listed below.

## 6.1. Function ID Table

The complete table of the Function ID used in command and response are listed in Appendix A.
It's recommended to have at least a one second delay between the **Setting Command** and the **Get Settings Command**.

## 6.2. Setting Command

The **Setting** command is a collection of many function-setting blocks in the following format:

**Command**
`<STX><S><FuncSETBLOCK1>…<FuncBLOCKn><ETX><LRC>`

**Response**
`<ACK>` or `<NAK>` for the wrong commands such as invalid funcID, length, and value.

Each function-setting block `<FuncSETBLOCK>` has the following format:
`<FuncID><Len><FuncData>`

**Where**:
`<FuncID>` is 1-byte identifying the setting(s) for the function.
`<Len>` is the 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.

## 6.3. Get Setting

The **Get Setting** command retrieves the reader's current settings.

**Command**

```
<STX> <R> <FuncID> <ETX> <LRC 1>
```

**Response**

```
<ACK> <STX> <FuncID> <Len> <FuncData> <ETX> <LRC 2>
```

`<FuncID>, <Len> and <FuncData>` retrieves the reader's current settings.

**Where**:

| Characters | Hex Value | Description |
|---|---|---|
| <STX> | 02 | Start of Text |
| <ETX> | 03 | End of Text |
| <ACK> | 06 | Acknowledge |
| <NAK> | 15 for RS232 and USB HID interface; FD for USB KB interface | Negative Acknowledge |
| <UnknownID> | 16 | **Warning**: Unsupported ID in se |
| <AlreadyInPOS> | 17 | **Warning**: Reader already in OP mode. |
| <R> | 52 | Review Setting |
| <S> | 53 | Send Setting |
| <LRC> | - | Xor'd all the data before LRC. |

**Reader Command Summary**

| ASCII | HEX | Name | Use |
|---|---|---|---|
| '8' | 38 | **Copyright Report** | Requests reader's copyright notice. |
| '9' | 39 | **Version Report** | Requests version string. |
| 'F" | 46 | **Key Loading** | Special command to load encryption keys. |
| 'I' | 49 | **Reader Reset** | Reset the reader. Software reset does not resend startup string |
| 'M' | 4D | **OPOS/ JPOS Command** | Command to enter **OPOS** or **JPOS** mode. |
| 'P' | 50 | **Arm and Disarm to Read** | **Arm to Capture Buffer Mode MSR**. |
| 'Q' | 51 | **Read Buffered Data** | **Read Stored MSR Data**. |
| 'R' | 52 | **Read MSR Options** | Read various reader optional settings. |
| 'S' | 53 | **Set MSR Options** | Set various reader optional functions. |

## 6.4. Get Copyright Information

A '31-byte' **Copyright Notice** will be returned:

```
02 38 03 39
```

**Response**:

```
ACK STX <Copyright String> ETX LRC
```

**Response mixed with Hex and ASCII**:

```
\06\02Copyright (c) 2010, ID TECH \03>
```

## 6.5. Version Report Command

```
02 39 03 38
```

**Response**:

```
ACK STX<Version String> ETX LRC
```

**Response mixed with Hex and ASCII**:

```
\06\02ID TECH TM3 SecureMag RS232 Reader V 3.19\03\LRC
```

## 6.6. Reader Reset Command

The reader supports a reset reader command and this command allows the host to return the reader to its default state.

```
02 49 03 48
```

**Response**:

```
06
```

## 6.7. OPOS/JPOS Command

There are three forms of the command:

`02 4D 01 30 03 7D` Enter Standard Mode (Exit **OPOS Mode**)

`02 4D 01 31 03 7C` Enter OPOS Mode

`02 4D 01 32 03 7F` Enter JPOS Mode

**Responses are as follows**:

`17` Reader already in **OPOS Mode**

`15` Command failure (wrong length or wrong parameter)

`06` Success

## 6.8. Arm or Disarm to Read Command

### 6.8.1. Arm to Read

This command enables the MSR to be ready for a card swipe in buffered mode.

- Any previously read data will be erased and reader will wait for the next swipe.
- As the user swipes a card, the data will be saved but not be sent to the host.
- The reader holds the data until receiving the next “Arm to Read” or “MSR Reset” command.

**Arm to Read**

```
02 50 01 30 03 LRC
```

### 6.8.2. Disarm to Read

This command will disable MSR read and clear any magnetic data in buffered mode. The reader enters to a disarmed state and will ignore MSR data.

Response is as follows:

| **Other possible response statuses**: | |
|---|---|
| NAK | 'P' command length must be 1 |
| NAK | 'P' command must be 0x30 or 0x32 |
| NAK | Reader not configured for buffered mode |
| NAK | Reader not configured for magstripe read |

NAK for keyboard interface is FD and non-KB mode NAK is `15`.

**Disarm to Read**

```
02 50 01 32 03 LRC
```

## 6.9. Read Buffered MSR Data Command

This command requests card data information for the buffered mode:

```
02 51 01 <Track Selection Option> 03 LRC
```

The `<Track Select Option>` byte:

| | |
|---|---|
| **‘0’** | Any Track |
| **‘1’** | Track 1 |
| **‘2’** | Track 2 |
| **‘3’** | Track 1 and Track 2 |
| **‘4’** | Track 3 |
| **‘5’** | Track 1 and Track 3 |
| **‘6’** | Track 2 and Track 3 |
| **‘7’** | Track 1, Track 2 and Track 3 |
| **‘8’** | Track 1 or Track 2 |
| **‘9’** | Track 2 or Track 3 |

The selected MSR data is sent to the host with or without envelope format, according to the operation mode setting.

This command does not erase the data.

**Response**:

```
06 02 <Len_H> <Len_L> <MSR Data> 03 LRC
```

Other possible response statuses:

- 18 'Q' command length must be 1
- 18 Reader not configured for buffered mode
- NAK Already armed

NAK for keyboard interface is FD, non-KB mode NAK is 15

## 6.10. Read MSR Options Command

```
02 52 1F 03 LRC
```

`<Response>` format:

The current setting data block is a collection of many function-setting blocks `<FuncSETBLOCK>` as follows:

`<STX><FuncSETBLOCK1>…<FuncSETBLOCKn><ETX><CheckSum>`

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

`<FuncID><Len><FuncData>`

**Where**:

- `<FuncID>` is 1-byte identifying the setting(s) for the function.
- `<Len>` is a 1-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.
- `<FuncSETBLOCK>` are in the order of their Function ID`<FuncID>`.

## 6.11. Set MSR Options Command

The default value is listed in **bold**.

### 6.11.1. Beep Volume

The beep volume and frequency can be each adjusted to two different levels or turned off.

```
02 53 11 01 <Beep Settings>03 LRC
```

Beep Settings:
‘0’ for beep volume off
‘1’ for beep volume high, low frequency
**‘2’ for beep volume high, high frequency**
‘3’ for beep volume low, high frequency
‘4’ for beep volume low, low frequency

Change to **Default Settings**:

```
02 53 18 03 LRC
```

This command does not have any `<FuncData>`. It returns all non-security settings for all groups to their default values.

# 7. MSR Reading Settings:

`02 53 1A 01<MSR Reading Settings> 03 LRC`

'0' MSR Reading Disabled
**'1' MSR Reading Enabled**

## 7.1. Decoding Method Settings

`02 53 1D 01<Decoding Method Settings> 03 LRC`

Decoding Method Settings:

- '0' Raw Data Decoding in Both Directions
- **'1' Decoding in Both Directions**
- '2' Moving stripe along head in direction of encoding
- '3' Moving stripe along head against direction of encoding

**Bi-Directional Method**: the user can swipe the card in either direction and the reader still processes the data encoded on the magnetic stripe.

**Raw Decoding**: sends the card's magnetic data in groups of 4 bits per character. There is no checking done except to verify track has or does not have magnetic data.

## 7.2. Terminator Setting

Terminator characters are used to end a string of data in some applications.
`02 53 21 01 <Terminator Settings> 03 LRC`

`<Terminator Settings>`
Any one character, `00h` is none; default is **CR** `(0Dh)`.

## 7.3. Preamble Setting

Characters can be added to the beginning of a string of data. These can be special characters for identifying a specific reading station, to format a message header expected by the receiving host, or any other character string.

Up to fifteen ASCII characters can be defined.

`02 53 D2 <Len><Preamble> 03 LRC`

**Where**:
Len = the number of bytes of Preamble string
Preamble = `{string length} {string}`

**Note**: String length is one byte, maximum fifteen `<0Fh>`.

## 7.4. Postamble Setting

The Postamble serves the same purpose as the Preamble, except it is added to the end of the data string, after any terminator characters.

```
02 53 D3 <Len><Postamble> 03 LRC
```

**Where**:
Len = the number of bytes of postamble string
Postamble = `{string length}{string}`

**Note:** String length is one byte, maximum fifteen <0Fh>.

## 7.5. Track n Prefix Setting

Characters can be added to the beginning of a track data. These can be special characters to identify the specific track to the receiving host, or any other character string. Up to six ASCII characters can be defined.

```
02 53 <n><Len><Prefix> 03 LRC
```

**Where**:
n is 34h for track 1; 35h for track 2 and 36h for track 3
Len = the number of bytes of prefix string
Prefix = `{string length}{string}`

**Note**: String length is one byte

## 7.6. Track x Suffix Setting

Characters can be added to the end of track data. These can be special characters to identify the specific track to the receiving host, or any other character string. Up to six ASCII characters can be defined.

```
02 53 <n><Len><Suffix> 03 LRC
```

**Where**:
n is 37h for track 1; 38h for track 2 and 39h for track 3
Len = the number of bytes of suffix string
Suffix = `{string length}{string}`

**Note**: **S**tring length is 1-byte, maximum six.

## 7.7. Track Selection

There are up to three tracks of encoded data on a magnetic stripe.
This option selects the tracks that will be read and decoded.

```
02 53 13 01 <Track_Selection Settings> 03 LRC
```

`<Track_Selection Settings>`

**'0' Any Track**
'1' Require Track 1 Only
'2' Require Track 2 Only
'3' Require Track 1 & Track 2
'4' Require Track 3 Only
'5' Require Track 1 & Track 3
'6' Require Track 2 & Track 3
'7' Require All Three Tracks
'8' Any Track 1 & 2
'9' Any Track 2 & 3

**Note**: If any of the required multiple tracks fail to read for any reason, no data for any track will be sent.

## 7.8. Track Separator Selection

Allows a user to select the character that separates data decoded by a multiple-track reader.

```
02 53 17 01 <Track_Separator> 03 LRC
```

`<Track_Separator>` is one ASCII Character.

**The default value is CR**, `0h` means no track separator.

## 7.9. Start and End Sentinel (Track 2 Account Number Only)

The SecureMag can be set to **Send** or **Not Send**, the **Start** or **End** sentinel, and to send either the Track 2 account number only, or all the encoded data on Track 2. (The Track 2 account number setting doesn't affect the output of Track 1 and Track 3.)

```
02 53 19 01 <SendOption> 03 LRC
```

`<SendOption>`

- '0' Don't send start and end sentinel and send all data on Track 2.
- **'1' Send start and end sentinel and send all data on Track 2.**
- '2' Don't send start and end sentinel and send account # on Track 2.
- '3' Send start and end sentinel and send account number on Track 2.

# 8. Security Features

The reader features configurable security settings. Before encryption can be enabled, **Key Serial Number** (**KSN**) and **Base Derivation Key** (BDK) must be loaded before encrypted transactions can take place. The keys are to be injected by certified key injection facility.

There are five **Security Level** available on the reader as specified in the followings:

| **Level 0** |
|---|
| Security Level 0 is a special case w**here** all DUKPT keys have been used and reset automatically when it runs out of DUKPT keys. The lifetime of DUKPT keys is 1 million. Once the key's end life is reached, the user should inject DUKPT keys again before doing any more transactions. |
| **Level 1** |
| By default, readers from the factory are configured to have this security level. There is no encryption process and no key serial number is transmitted with decoded data. The reader functions as a non-encrypting reader and the decoded track data is sent out in default mode. |
| **Level 2** |
| **Key Serial Number** and **Base Derivation Key** have been injected but the encryption process is not yet activated. The reader will send out decoded track data in default format. Setting the encryption type to TDES and AES will change the reader to Security Level 3. |
| **Level 3** |
| Both the **Key Serial Number** and **Base Derivation Keys** are injected and then **Encryption Mode** is turned on. For payment cards, both encrypted data and masked clear text data are sent out. Users can select the data masking of the PAN area; the encrypted data format cannot be modified. Users can choose whether to send hashed data and whether to reveal the card expiration date. |
| **Level 4** |
| When the reader is at Security Level 4, a correctly executed **Authentication Sequence** is required before the reader sends out data for a card swipe. Commands that require security must be sent with a 4-byte **Message Authentication Code** (MAC) at the end. |

**Note**: Data supplied to MAC algorithm should NOT be converted to ASCII-Hex, rather it should be supplied in its raw binary form. Calculating MAC requires knowledge of current DUKPT **KSN** which can be retrieved with the Get **DUKPT KSN** and **Counte**r command.

Default reader properties are configured to have Security Level 1 (no encryption). In order to output encrypted data, the key must be injected into the reader while the encryption feature enabled. The reader will configure to Security Level 2, 3 or 4 and it cannot be reverted to a lower security level.

## 8.1. Encryption Management

The Encrypted swipe read supports TDES and AES encryption standards for data encryption. Encryption can be turned on via a command. TDES is the default.

If the reader is in Security Level 3, for the encrypted fields, the original data is encrypted using the **TDES/AES CBC** mode with an **Initialization Vector** starting at all binary zeroes and the **Encryption Key** associated with the current **DUKPT KSN**.

## 8.2. Check Card Format

ISO/ABA (American Banking Association) Card (card type 0)
**Encoding Method**:

- Track1 is 7 bits encoding.
- Track1 is 7 bits encoding.
- Track2 is 5 bits encoding.
- Track3 is 5 bits encoding.
- Track1 is 7 bits encoding.
- Track2 is 5 bits encoding.
- Track2 is 5 bits encoding.

Additional check

- Track1 2nd byte is 'B'.
- There is only one '=' in track 2 and the position of '=' is between 13th ~ 20th character so account number length is 12-19 digits.
- Total length of track 2 is above 19 characters.

AAMVA (American Association of Motor Vehicle Administration) Card
Encoding method:

- Track1 is 7 bits encoding. Track2 is 5 bits encoding. Track3 is 7 bits encoding.
- Others (Customer card)

## 8.3. MSR Data Market

For ABA Card Data (Card Type 0)
For cards need to be encrypted, both encrypted data and clear text data are sent.

Masked Area

- The data format of each masked track is ASCII.
- The clear data include start and end sentinels, separators, first N, last M digits of the PAN, card holder name (for Track1).

The rest of the characters should be masked using mask character.
`Set PrePANClrData (N), PostPANClrData (M), MaskChar (Mask Character)`
N and M are configurable and default to 4 first and 4 last digits. They follow the current PCI constraints requirements (N 6, M 4 maximum).

Mask character default value is '*'.

- Set PrePANClrDataID (N), parameter range `00h ~ 06h,` default value `04h`

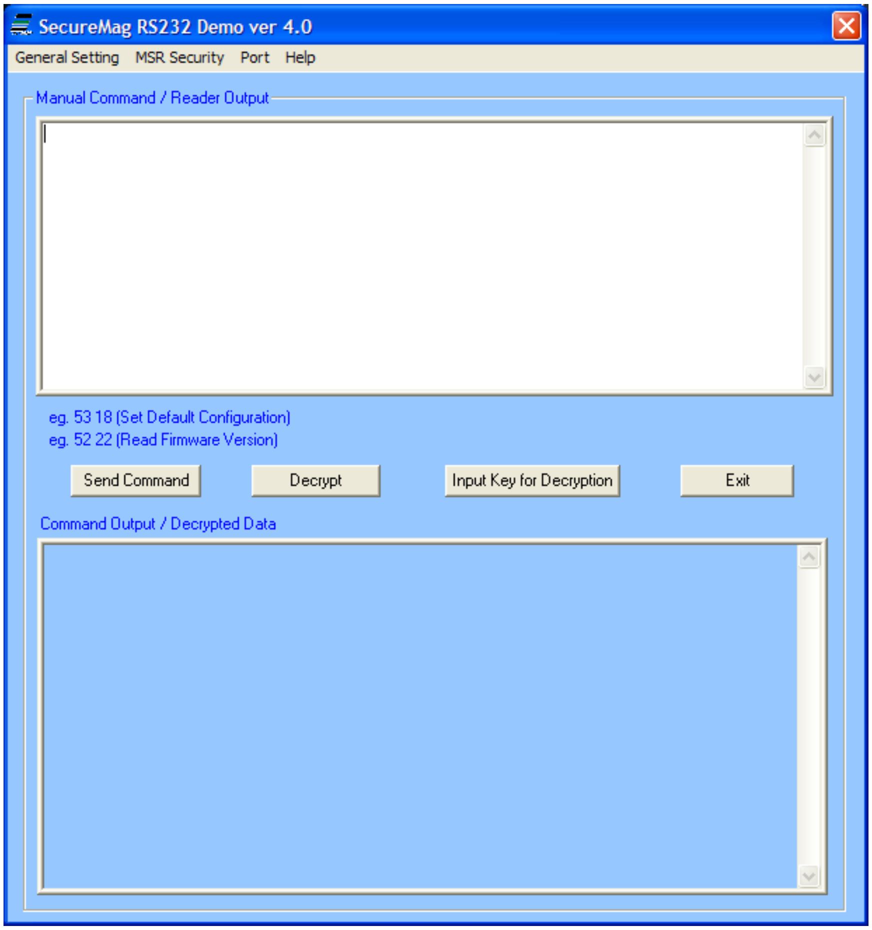
- Set PostPANClrDataID (M), parameter range 00h ~ 04h, default value 04h
- MaskCharID (Mask Character), parameter range 20h ~ 7Eh, default value 2Ah
- DisplayExpirationDataID, parameter range ‘0’~’1’, default value ‘0’

# 9. Demo Program

ID TECH SecureMag Demo is provided to demonstrate features of the Encrypted MSR. It supports decrypting the encrypted data and sending command to MSR.

## 9.1. Overview of SecureMag Demo

The demo software is similar for each interface with exception of interface- specific settings.


SecureMag RS232 Demo ver 4.0
General Setting
MSR Security
Port
Help
Manual Command / Reader Output
eg. 53 18 (Set Default Configuration)
eg. 52 22 (Read Firmware Version)
Send Command
Decrypt
Input Key for Decryption
Exit
Command Output / Decrypted Data

### 9.1.1. Manual Command

The demo software allows users to manually input and send commands.

Type <Command Data> in the field and the command send in the below format:
<STX> <Command_Data> <ETX> <LRC>

**Where**:

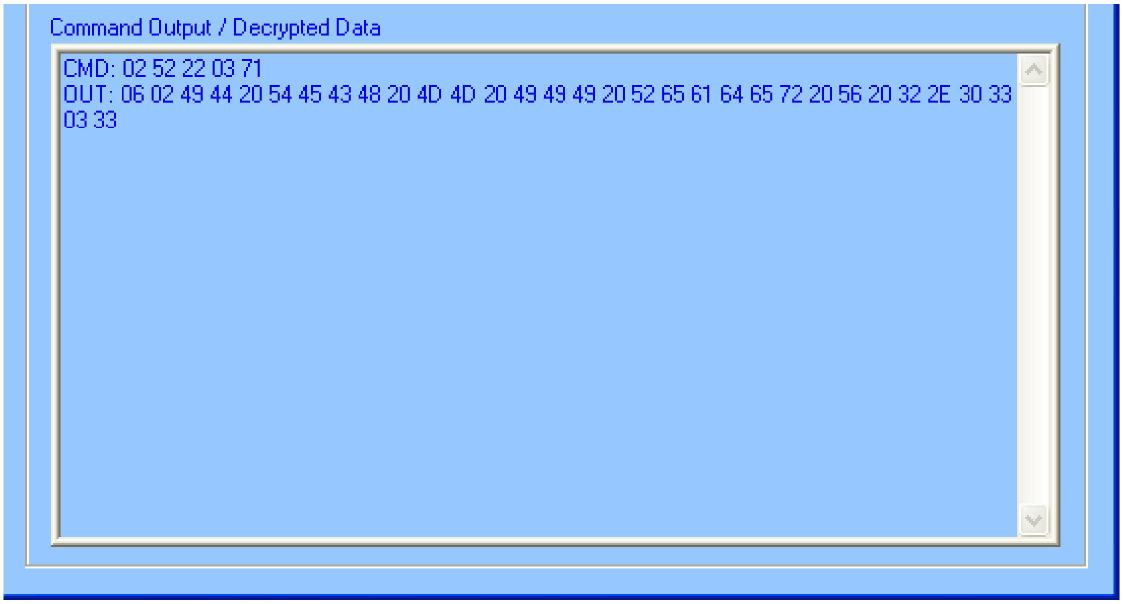
<STX> = 02h, <ETX> = 03h.

<Command_Data>: Please refer to Appendix A for a complete list of commands

`<LRC>` is a 1-byte Xor value calculated for the above data block from `<STX>` to `<ETX>`.

For example, `02 53 18 03 4A`, **Set Default Configuration**.
For example, `02 52 22 03 71`, **Read Firmware Version.**

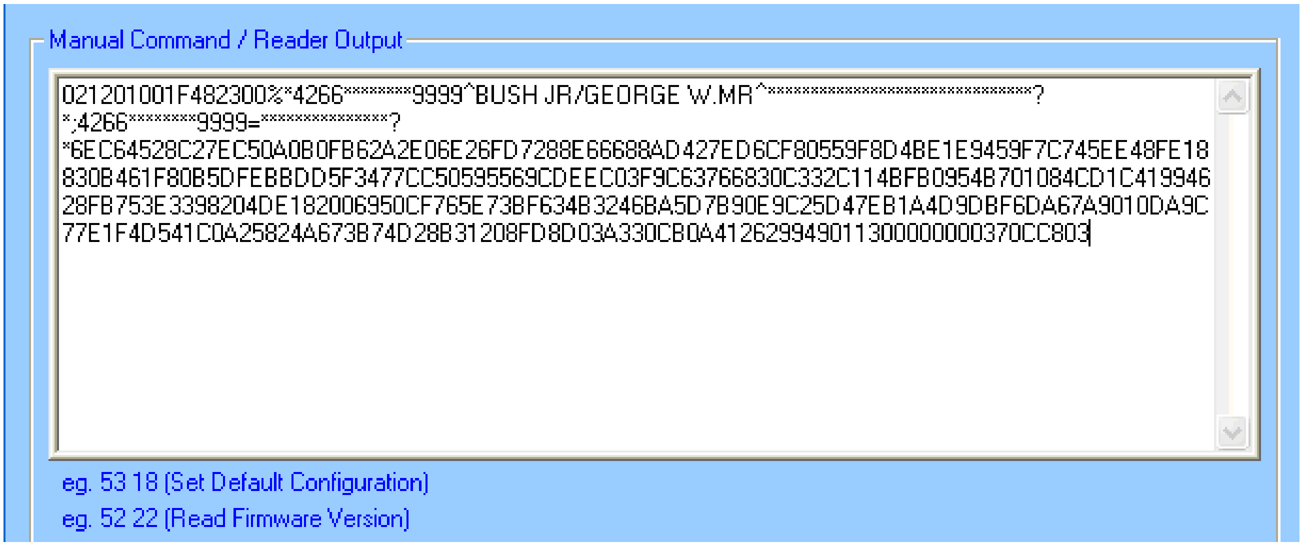
Click **Send Command**, and the input and output would be shown in the lower text box.



```
CMD: 02 52 22 03 71
OUT: 06 02 49 44 20 54 45 43 48 20 4D 4D 20 49 49 49 20 52 65 61 64 65 72 20 56 20 32 2E 30 33
03 33
```

## 9.2. Decryption

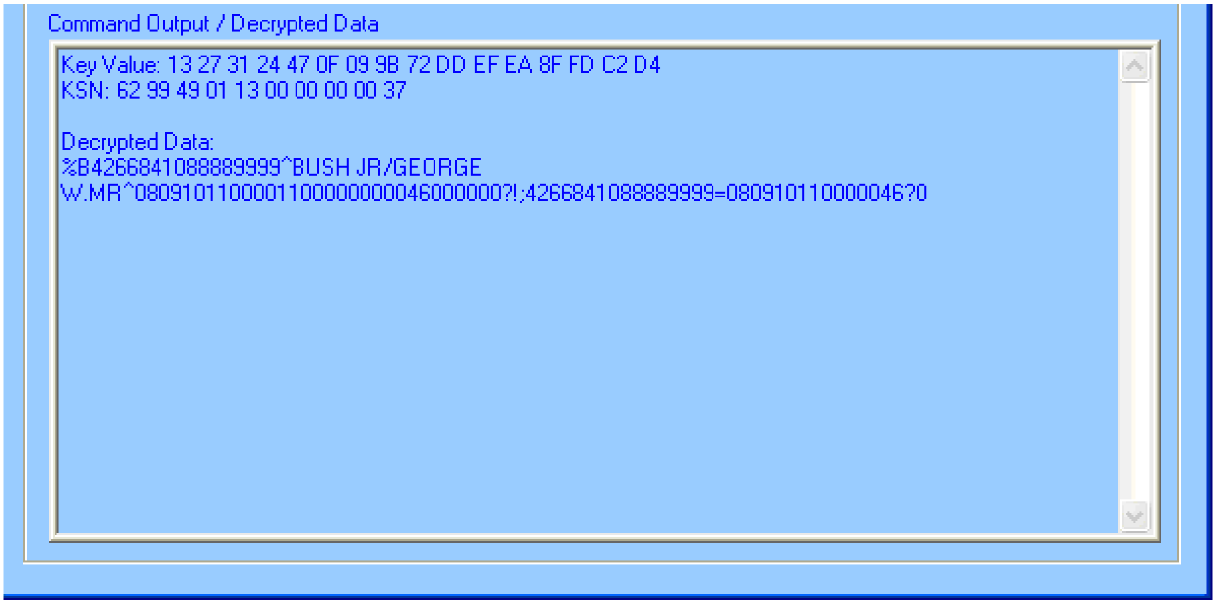
The encrypted data will show in the **Manual Command / Encrypted Data** textbox after a card is swiped. By default, the cursor is in **Manual Command / Encrypted Data** textbox.



```
021201001F482300%*4266********9999^BUSH JR/GEORGE W.MR^********************************?
*;4266********9999=***************?
*6EC64528C27EC50A0B0FB62A2E06E26FD7288E66688AD427ED6CF80559F8D4BE1E9459F7C745EE48FE18
830B461F80B5DFEBBDD5F3477CC50595569CDEEC03F9C63766830C332C114BFB0954B701084CD1C419946
28FB753E3398204DE182006950CF765E73BF634B3246BA5D7B90E9C25D47EB1A4D9DBF6DA67A9010DA9C
77E1F4D541C0A25824A673B74D28B31208FD8D03A330CB0A41262994901130000000370CC803
```

eg. 53 18 (Set Default Configuration)
eg. 52 22 (Read Firmware Version)

Click the **Decrypt** button and the decrypted card data will be displayed in the lower box.



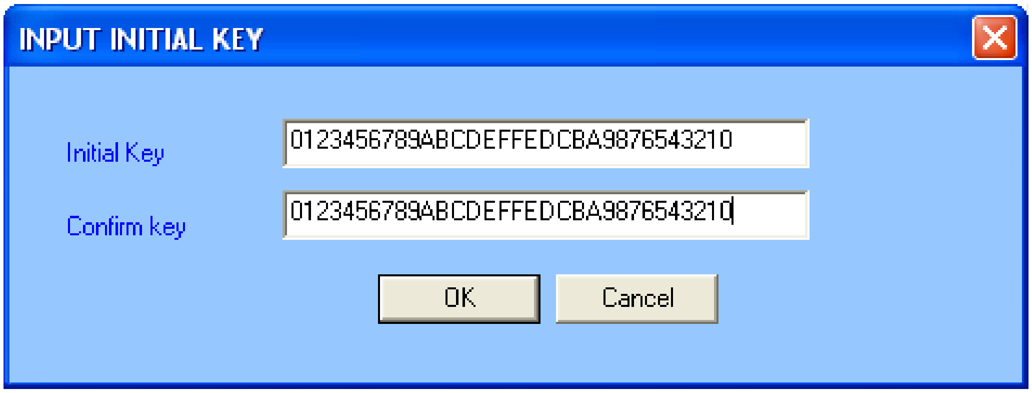
Command Output / Decrypted Data

```
Key Value: 13 27 31 24 47 0F 09 9B 72 DD EF EA 8F FD C2 D4
KSN: 62 99 49 01 13 00 00 00 00 37

Decrypted Data:
%B4266841088889999^BUSH JR/GEORGE
W.MR^0809101100001100000000046000000?!;4266841088889999=080910110000046?0
```

The default initial key is `0123456789ABCDEFFEDCBA9876543210`.

1. Click **Input Initial Key** to load the key into demo software**.** (Only if reader is programmed with the user-defined key.)
2. Re-type the key into the **Confirm Key** text box.
3. Click **Ok**.

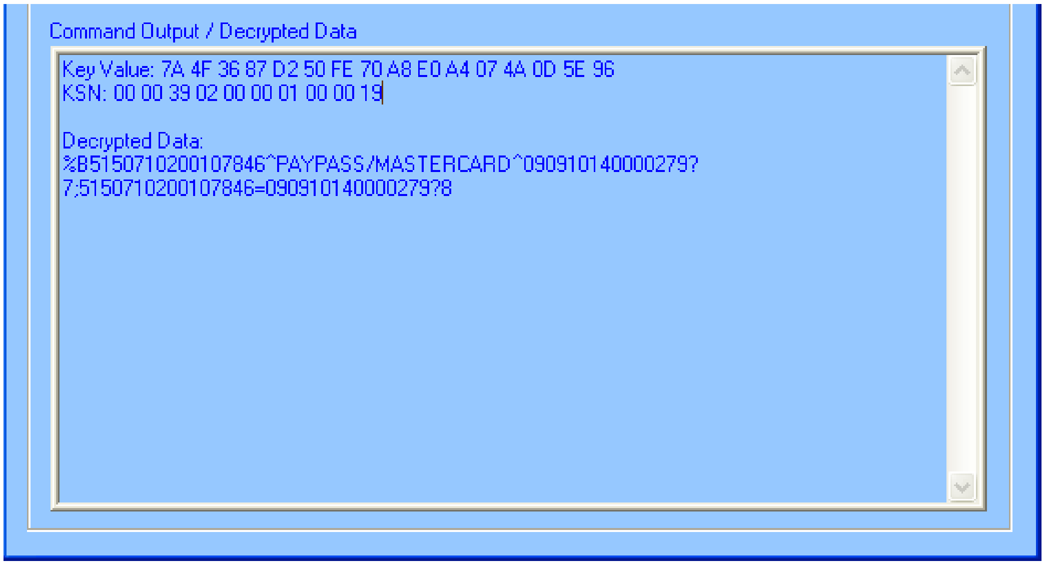


INPUT INITIAL KEY

Initial Key: 0123456789ABCDEFFEDCBA9876543210

Confirm key: 0123456789ABCDEFFEDCBA9876543210

The **Key Value**, **KSN**, and **Decrypted Data** will display in the **Command Output/ Decrypted Data** textbox.



```
Key Value: 7A 4F 36 87 D2 50 FE 70 A8 E0 A4 07 4A 0D 5E 96
KSN: 00 00 39 02 00 00 01 00 00 19

Decrypted Data:
%B5150710200107846^PAYPASS/MASTERCARD^090910140000279?
7;5150710200107846=09091014000027978
```

## 9.3. Reader Operations

The demo software can be used to display the card data and send reader commands. To view the card data on screen, place the cursor in the **Manual Command/ Reader Output** text box and swipe the card. To send a reader command, type the appropriate command in the text box and press the **Send Command** button.

- General Settings
  Provides options such as reader default settings, firmware version, beeper options, and buffered mode options. For USB demo software, there are options to set the reader to USB KB or USB HID mode.
- MSR Security
  The security is enabled by selecting TDES or AES. Once the encryption is enabled the reader cannot be changed back to non-encrypted mode.
- Port and Settings
  **RS232 Interface**: select Com port, open and close port.
  **USB KB Interface**: set KB polling interval and select language settings
- Help
  Provides version information of the demo software.

# 10. Data Format

The USB version of the reader can be operated in two different modes:

- HID ID TECH mode (**HID Mode**), Product ID: 2010
- HID with Keyboard Emulation (**KB Mode**), Product ID: 2030

When the reader is operated in the HID mode, it behaves like a vendor defined HID device. A direct communication path can be established between the host application and the reader without interference from other HID devices.

## 10.1. Level One and Level Two Standard Mode Output Format

**USB HID Output Format**

Card data is only sent to the host on the **Interrupt In** pipe using an **Input Report**. The reader will send one **Input Report** per card swipe. If the host requests data from the reader when no data is available, the reader will send a NAK to the host to indicate it has nothing to send.

### 10.1.1. USB HID Data Format

Other Mode Reader Data Structure

| Offset | Usage Name |
|---|---|
| 0 | T1 decode status |
| 1 | T2 decode status |
| 2 | T3 decode status |
| 3 | T1 data length |
| 4 | T2 data length |
| 5 | T3 data length |
| 6 | Card encode type |
| 7-116 | T1 data |
| 117-226 | T2 data |
| 227-336 | T3 data |

**Notes**:

- T1, T2 or T3 decode status: 0 for no error, 1 for error
- T1, T2 or T3 Data Length: Each byte value indicates how many bytes of decoded card data are in the track data field. This value will be zero if there was no data on the track or if there was an error decoding the track.

**Card Encode Type**:

| Value | Encode Type | Description |
|---|---|---|
| 0 | ISO/ABA | ISO/ABA encode format |
| 1 | AAMVA | AAMVA encode format |
| 3 | Other | The card has a non-standard format. For example, ISO/ABA track 1 format on track 2. |
| 4 | Raw | The card data is sent in a Raw encrypted format. All tracks are encrypted, and no mask data is sent. |

T1, T2 or T3 data: The length of each track data field is fixed at 110 bytes, but the length of valid data in each field is determined by the track data length field that corresponds to the track number. The track data includes all data string starting with the start sentinel and ending with the end sentinel.

ID TECH Reader Data Structure

| Offset | Usage |
|---|---|
| 0 | T1 decode status |
| 1 | T2 decode status |
| 2 | T3 decode status |
| 3 | T1 data length |
| 4 | T2 data length |
| 5 | T3 data length |
| 6 | Card encode type |

7,8 Total Output Length
9-HIDSIZE* Output Data

In this approach, the reader will keep all the ID TECH data editing and features like Preamble, Postamble, and other data.  The output data is HIDSIZE* bytes; the **Total Output Length** field indicates the valid data length in the output data.

**Note***: HIDSIZE (560 bytes as described in USB enumeration.) HIDSIZE is subject to change. Software should auto adjust in case enumeration changes.

### 10.1.2. Descriptor Tables

**Device Descriptor**:

| Field | Value | Description |
|---|---|---|
| **Length** | 12 | |
| **Des Type** | 01 | |
| **bcd USB** | 00 02 | USB 2.0 |
| **Device Class** | 00 | Unused |
| **Sub Class** | 00 | Unused |
| **Device Protocol** | 00 | Unused |
| **Max Packet Size** | 08 | |
| **VID** | 0A CD | |
| **PID** | 20 10<br>20 20<br>20 30 | HID ID TECH Structure<br>HID Other Structure<br>HID Keyboard |
| **BCD Device Release** | 00 01 | |
| **i-Manufacture** | 01 | |
| **i-Product** | 02 | |
| **i-Serial Number** | 00 | |
| **#Configuration** | 01 | |

**Configuration Descriptor**:

| Field | Value | Description |
|---|---|---|
| **Length** | 09 | |
| **Des type** | 02 | |
| **Total Length** | 22 00 | |
| **No. Interface** | 01 | |
| **Configuration Value** | 01 | |
| **iConfiguration** | 00 | |
| **Attributes** | 80 | Bus power, no remove wakeup |
| **Power** | 32 | 100 mA |

**Interface Descriptor**:

| Field | Value | Description |
|---|---|---|
| **Length** | 09 | |
| **Des Type** | 04 | |
| **Interface No.** | 00 | |
| **Alternate Setting** | 00 | |
| **#EP** | 01 | |
| **Interface Class** | 03 | |
| **Sub Class** | 01 | HID |
| **Interface Protocol** | 01 | |
| **iInterface** | 00 | |

**HID Descriptor**:

| Field | Value | Description |
|---:|---|---|
| **Length** | 09 | |
| **Des Type** | 21 | HID |
| **bscHID** | 11 01 | |
| **Control Code** | 00 | |
| **numDescriptors** | 01 | Number of Class Descriptors to follow. |
| **DescriptorType** | 22 | Reporter Descriptor |
| **Descriptor Length** | 37 00<br>3D 00<br>52 00 | HID ID Tech Format<br>HID Other Format<br>HID Keyboard Format |

**End Pointer Descriptor**:

| Field | Value | Description |
|---:|---|---|
| **Length** | 07 | |
| **Des Type** | 05 | End Point |
| **EP Addr** | 83 | EP3-Im |
| **Attributes** | 03 | Interrupt |
| **MaxPacketSize** | 40 00 | |
| **bInteraval** | 01 | |

**Report Descriptor**: (USB-HID Setting)

| Value | Description |
|---:|---|
| 06 00 FF | Usage Page (MSR) |
| 09 01 | Usage (Decode Reader Device) |
| A1 01 | Collection (Application) |
| 15 00 | Logical Minimum |
| 26 FF 00 | Logical Maximum |
| 75 08 | Report Size |
| 09 20 | Usage (Tk1 Decode Status) |
| 09 21 | Usage (Tk2 Decode Status) |
| 09 22 | Usage (Tk3 Decode Status) |
| 09 28 | Usage (Tk1 Data Length) |
| 09 29 | Usage (Tk2 Data Length) |
| 09 2A | Usage (Tk3 Data Length) |
| 09 38 | Usage (Card Encode Type) |
| 95 07 | Report Count |
| 81 02 | Input (Data, Var, Abs, Bit Field) |
| 09 30 | Usage (Total Sending Length) |
| 95 02 | Report Count (2) |
| 82 02 01 | Input (Data, Var, Abs, Bit Field) |
| 09 31 | Usage (Output Data) |
| 96 27 02 | Report Count (HIDSIZE = 551+9=560 ) |
| 82 02 01 | Input (Data, Var, Abs, Bit Field) |
| 09 20 | Usage (Command Message) |

| | |
|---|---|
| 95 08 | Report Count |
| B2 02 01 | Feature (Data,Var, Abs, Buffered Bytes) |
| C0 | End Collection |

**Reader Descriptors**:

| Value | Description |
|---|---|
| 05 01 | Usage Page (Generic Desktop) |
| 09 06 | Usage (Keyboard) |
| A1 01 | Collection (Application) |
| 05 07 | Usage Page (Key Codes) |
| 19 E0 | Usage Minimum |
| 29 E7 | Usage Maximum |
| 15 00 | Logical Minimum |
| 25 01 | Logical Maximum |
| 75 01 | Report Size |
| 95 08 | Report Count |
| 81 02 | Input (Data, Variable, Absolute) |
| 95 01 | Report Count (1) |
| 75 08 | Report Size |
| 81 01 | Input Constant |
| 95 05 | Report Count |
| 75 01 | Report Size |
| 05 08 | Usage Page (LED) |
| 19 01 | Usage Minimum |
| 29 05 | Usage maximum |
| 91 02 | Output (Data Variable Absolute) |
| 95 01 | Report Count |
| 75 03 | Report Size |
| 91 01 | Output (Constant) |
| 95 06 | Report Count |
| 75 08 | Report Size |
| 15 00 | Logical Minimum |
| 25 66 | Logical Maximum (102) |
| 05 07 | Usage Page (key Code) |
| 19 00 | Usage Minimum |
| 29 66 | Usage Maximum (102) |
| 81 00 | Input(Data, Array) |
| 06 2D FF | Usage Page (ID TECH) |
| 95 01 | Report Count |
| 26 FF 00 | Logical maximum (255) |
| 15 01 | Logical Minimum |
| 75 08 | Report Size (8) |
| 09 20 | Usage (Setup data byte) |
| 95 08 | Report Count (8) |
| B2 02 01 | Feature (Data Var, Abs) |
| C0 | End Collection |

## 10.2. Level One and Level Two POS Mode Data Output Format

In POS mode, use the special envelope to send out card data, envelope is in the following format: `[Right Shift, Left Shift, Right Ctrl, Left Ctrl,] Read Error, Track x ID; Track x Error; Track x Data Length; Track x Data; Card Track x LEC code; Track x data LRC.`

Reader will send out card data in Alt mode if its ASCII code less than H'20'.

| Byte No. | Name |
|---:|---|
| **0** | Right Shift |
| **1** | Left Shift |
| **2** | Right Ctrl |
| **3** | Left Ctrl |
| **4** | Read error 1 |
| **5** | Read error 2 |
| **6** | Track x ID |
| **7** | Track x Error |
| **8** | Track x Length 1 |
| **9** | Track x Length 2 |
| **10** | Track Data (no extra track ID for raw data) |
| | … |
| **10 + Track len -1** | Card Track x LRC |
| **10 + Track len** | Track x LRC |
| **10 + Track len +1** | 0 x 0D |
| **10 + Track len + 2** | Track x ID |
| **....** | Repeat Track |

The data format is independent with MSR setting. No Track x data if track x sampling data does not exist.

**OPOS header**:

Only HID KB interface has [`Right Shift, Left Shift, Right Ctrl, Left Ctrl`] under POS mode.

**Read Error**:

Read Error 1-byte bits:

| MB | | | | | | LB | |
|---|---|---|---|---|---|---|---|
| **0** | **B6** | **B5** | **B4** | **B3** | **B2** | **B1** | **B0** |
| **B0** | 1: Track 1 sampling data exists (0: Track 1 sampling data does not exist) | | | | | | |
| **B1** | 1: Track 2 sampling data exists (0: Track 2 sampling data does not exist) | | | | | | |
| **B2** | 1: Track 3 sampling data exists (0: Track 3 sampling data does not exist) | | | | | | |
| **B3** | 1: Track 1 decode success (0: Track 1 decode fail) | | | | | | |
| **B4** | 1: Track 2 decode success (0: Track 2 decode fail) | | | | | | |
| **B5** | 1: Track 3 decode success (0: Track 3 decode fail) | | | | | | |
| **B6** | 0: if b0 to b5 are all 1, otherwise 1 (make it printable) | | | | | | |

**Read Error byte 2**:

| MB | | | | | | LB | |
|---|---|---|---|---|---|---|---|
| **0** | **1** | **B12** | **B11** | **B10** | **B9** | **B8** | **B7** |
| B7 | 1: Track 4 sampling data exists (0: Track 4 sampling data does not exist) | | | | | | |
| B8 | 1: Track 4 JIS II decode success (0: Track4 JIS II decode fail) | | | | | | |
| B9, B10, B11 | | | | | | | |
| | | 000: ISO Card (7, 5) or (7, 5, 5) encoding | | | | | |
| | | 001: Old CADL Card (6, 5, 6) encoding (no longer included) | | | | | |
| | | 010: AAMVA Card (7, 5, 7) encoding | | | | | |
| | | 011: JIS I Card (8, 5, 8) encoding | | | | | |
| | | 100: JIS II card (8) or ISO+JIS II | | | | | |
| | | 110: OPOS Raw Data Output | | | | | |
| | | 111: JIS I + JIS II | | | | | |
| B12 | Reserved for future use. | | | | | | |

### Track ID

Track ID is a byte of ID, it will be `'1'`, `'2'`, and `'3'` for track 1, 2, and 3; it is not accurate to use start sentinel to identify track.

### Track x Error

Track x error is a byte of flags, it will be in format of: `0 0 1 b4, b3, b2 b1 b0`

| | |
|---|---|
| **b0** | 1: Start sentinel error (0: Not start sentinel error) |
| **b1** | 1: End sentinel error (0: Not end sentinel error) |
| **b2** | 1: Parity error (0: Not parity error) |
| **b3** | 1: LRC error (0: Not LRC error) |
| **b4** | 1: Other error (0: Not other error) |

Track x Error is set to 0x20 in OPOS raw data mode.

### Track Length

Assume actual "Track x Data Length" is hex code xy; the Track x data length for OPOS mode output will be Hex code 3x, 3y.

Track x data length does not include the byte of "`Track x data LRC`", it is `<30> <30>` in case of read error on track x.

**Track Data**
"`Card Track x LRC code`" is track x card data.

**Track x LRC**
"`Track x data LRC`" is an LRC to check track x data communication; XOR all characters start from "`Track x ID`" to "`Track x data LRC`" should be 0.

## 10.3. DUKPT Level Four Data Output Original Format

For ISO cards both masked clear, encrypted data, and no clear data are sent. Clear data is sent will be sent to other cards.

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 2-byte length of `<Card Data>`.
`<CheckLRC>` is a 1-byte **Exclusive-OR** sum calculated for all `<Card Data>`.
`<CheckSum>` is a 1-byte **Sum** value calculated for all `<Card data>`.
`<Card Data>` card data format is shown below.

**ISO/ABA Data Output Format**:

| | |
|---|---|
| card encoding type | (0: ISO/ABA, 4: for **Raw Mode**) |
| track status | (bit 0,1,2:T1,2,3 decode, bit 3,4,5:T1,2,3 sampling) |
| track 1 unencrypted length | (1-byte, 0 for no track1 data) |
| track 2 unencrypted length | (1-byte, 0 for no track2 data) |
| track 3 unencrypted length | (1-byte, 0 for no track3 data) |
| track 1 masked | (Omitted if in Raw mode) |
| track 2 masked | (Omitted if in Raw mode) |
| track 3 data | (Omitted if in Raw mode) |
| track 1,2 encrypted | (AES/TDES encrypted data) |
| track 1 hashed | (20 bytes SHA1-Xor) |
| track 2 hashed | (20 bytes SHA1-Xor) |
| DUKPT serial number | (10 bytes) |

**Non-ISO/ABA Data Output Format**:

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

## 10.4. Level 4 Data Output Original 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 2-byte length of `<Card Data>`.

`<CheckLRC>` is a 1-byte Exclusive-OR sum calculated for all `<Card Data>`.

`<CheckSum>` is a 1-byte Sum value calculated for all `<Card data>`.

`<Card Data>` format is

**ISO/ABA Data Output Format**:

- card encoding type (0: ISO/ABA, 4: for Raw Mode)
- track status (bit 0,1,2: T1,2,3 decode, bit 3,4,5:T1,2,3 sampling)
- track 1 unencrypted length (1-byte, 0 for no track1 data)
- track 2 unencrypted length (1-byte, 0 for no track2 data)
- track 3 unencrypted length (1-byte, 0 for no track3 data)
- If card encoding type high bit set
    - mask and clear sent track status
    - encrypt and hash sent track status

**In this mode tracks are encrypted separately rather than as a group**:

- track 1 masked (Omitted if in Raw mode)
- track 2 masked (Omitted if in Raw mode)
- track 3 data (Omitted if in Raw mode)

| | |
|---|---|
| track 1&2 encrypted | (AES/TDES encrypted data) |
| sessionID encrypted | (AES/TDES encrypted data) |
| track 1 hashed | (20-bytes SHA1-Xor) |
| track 2 hashed | (20-bytes SHA1-Xor) |
| track 3 hashed (optional) | (20-bytes SHA1-Xor) |
| DUKPT serial number | (10-bytes) |

**Non-ISO/ABA Data Output Format**:

| | |
|---|---|
| card encoding type | (1: AAMVA, 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, 0 for no track1 data) |
| track 2 length | (1-byte, 0 for no track2 data) |
| track 3 length | (1-byte, 0 for no track3 data) |
| track 1 data | |
| track 2 data | |
| track 3 data | |

## 10.5. DUKPT Level 3 Data Output Enhanced Format

This format is the standard encryption format but not yet the default encryption format.

This mode is used for the following reasons below:

- When all tracks must be encrypted.
- When encrypted OPOS support is required.
- When the tracks must be encrypted separately.
- When cards other than type 0 (ABA bank cards) must be encrypted.
- When track 3 must be encrypted.

**1. <u>Encryption Output Format Setting</u>**:

**Command**:

```
53 85 01 <Encryption Format>
```

**Encryption Format**:

**'00h': Original Encryption Format**

'01h': Enhanced Encryption Format

**2. <u>Encryption Option Setting</u>**: (for enhanced encryption format only)

**Command**:

```
 53 84 01 <Encryption Option>
```

Encryption Option: (**default 08h**)

bit0: 1 – track 1 force encrypt

bit1: 1 – track 2 force encrypt
bit2: 1 – track 3 force encrypt
bit3: 1 – track 3 force encrypt when card type is 0
bit4: 1 – new mask feature: see note 4) below

**Note**:

1. When force encrypt is set, this track will always be encrypted, regardless of card type. No clear/mask text will be sent.
2. If and only if in enhanced encryption format, each track is encrypted separately. Encrypted data length will round up to 8 or 16 bytes.
3. When force encrypt is not set, the data will be encrypted in original encryption format, that is, only track 1 and track 2 of type 0 cards (ABA bank cards) will be encrypted.
4. When new mask feature (bit 4) is set Mask data can be sent even if set to "force encrypt" (bit0-3 is set); bIf bank card and track 3 is ISO-4909 with PAN format, T3 will be encrypted and has mask data.

**Typical Settings**:

1. **08** (default):
   **Bank card**: All three tracks will be encrypted. Only T1 and T2 can have mask.
   **Non-Bank card**: Will be sent in clear text.
2. **07**
   **Force encryption**: All three tracks will be encrypted without mask, regardless of card type.

3. **10**
   **Bank card**: T1 and T2 will be encrypted. If the T3 is with ISO-4909 format, it'll be encrypted and its mask data will be sent out. Otherwise, T3 will be sent in clear text.
   **Non-Bank card:** Will be sent in clear text.
4. **17**
   **Bank card**: All three tracks will be encrypted. T3 will allow to send mask if it's ISO-4909 format.
   **Non-Bank card**: Will be encrypted without mask.

### 3. Hash Option Setting:

**Command**: `53 5C 01 <Hash Option>`

| Hash Option: ('0' – '7') | |
|---|---|
| **Bit0:** | 1 – track1 hash will be sent if data is encrypted |
| **Bit1:** | 1 – track2 hash will be sent if data is encrypted |
| **Bit2:** | 1 – track3 hash will be sent if data is encrypted |

### 4. Mask Option Setting: (for enhanced encryption format only)

**Command**:

```
53 86 01 <Mask Option>
```

| **Mask Option: (Default: 0x07)** | |
|---|---|
| **Bit0:** | 1 – tk1 mask data allow to send when encrypted |
| **Bit1:** | 1 – tk2 mask data allow to send when encrypted |
| **Bit2:** | 1 – tk3 mask data allow to send when encrypted |

When mask option bit is set – if data is encrypted (but not forced encrypted), the mask data will be sent; If mask option is not set, the mask data will not be sent under the same condition.

**Settings for OPOS**:

- Assume reader is under default setting (Encrypt Structure 0)
- Set to new Encrypt Structure 1: `53 85 01 31`

The OPOS driver and application may also send following command when changing (**Decode or Raw** format)

(Set raw or decode data format)

`53 1D 01 30` // RAW data format

`53 1D 01 31` // Decoded format

Card data is sent out in the following format:

```
<STX><LenL><LenH><Card Data><CheckLRC><CheckSum><ETX>
Where <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>` card data format is shown below.

**ISO/ABA Data Output Format:**

| | |
|---|---|
| **0** | STX |
| **1** | Data Length low byte |
| **2** | Data Length high byte |
| **3** | Card Encode Type1 |
| **4** | Track 1-3 Status2 |
| **5** | Track 1 unencrypted data length |
| **6** | Track 2 unencrypted data length |
| **7** | Track 3 unencrypted data length |
| **8** | Clear/masked data sent status 3 |
| **9** | Encrypted/Hash data sent status 4 |
| **10** | Track 1 clear/mask data |

| | |
|---|---|
| Track 2 | clear/mask data |
| Track 3 | clear/mask data |
| Track 1 | encrypted data |
| Track 2 | encrypted data |
| Track 3 | encrypted data |
| Session ID (8 bytes) (Security Level 4 only) | |

| | |
|---|---|
| Track 1 | hashed (20 bytes each) (if encrypted and hash track 1 allowed) |
| Track 2 | hashed (20 bytes each) (if encrypted and hash track 2 allowed) |
| Track 3 | hashed (20 bytes each) (if encrypted and hash track 3 allowed) |

**KSN** (10 bytes)

**CheckLRC**

**CheckSum**

**ETX**

| **Non-ISO/ABA Data Output Format:** | |
|---|---|
| **0** | STX |
| **1** | Data Length low byte |
| **2** | Data Length high byte |
| **3** | Card Encode Type* |
| **4** | Track 1-3 Status |
| **5** | T1 unencrypted data length |
| **6** | T2 unencrypted data length |
| **7** | T3 unencrypted data length |
| **8** | Clear/mask data sent status * |
| **9** | Encrypted/Hash data sent status * |
| **10** | T1 clear data<br>T2 clear data<br>T3 clear data<br>CheckLrc<br>CheckSum |

**Note 1**: Card Encode Type

Card Type will be 8x for enhanced encryption format and 0x for original encryption format

| Value | Encode Type Description |
|---|---|
| 00h / 80h | ISO/ABA format |
| 01h / 81h | AAMVA format |
| 03h / 83h | Other |
| 04h / 84h | Raw; un-decoded format |

For Type 04 or 84 Raw data format, all tracks are encrypted, and no mask data is sent. No track indicator ‘01’, ‘02’ or ‘03’ in front of each track. Track indicator ‘01’,’02’ and ‘03’ will still exist for non-encrypted mode.

**Note 2**: Track 1-3 status byte.

| Field 4: | |
|---|---|
| Bit 0: | 1— track 1 decoded data present |
| Bit 1: | 1— track 2 decoded data present |
| Bit 2: | 1— track 3 decoded data present |
| Bit 3: | 1— track 1 sampling data present |
| Bit 4: | 1— track 2 sampling data present |
| Bit 5: | 1— track 3 sampling data present |
| Bit | 6, 7 — Reserved for future use |

**Note 3**: Clear/mask data sent status.

Field 8 (Clear/mask data sent status) and field 9 (Encrypted/Hash data sent status) will only be sent out in enhanced encryption format.

| Field 8: Clear/masked data sent status byte: | |
|---|---|
| Bit 0: | 1 —track 1 clear/mask data present |
| Bit 1: | 1— track 2 clear/mask data present |
| Bit 2: | 1— track 3 clear/mask data present |
| Bit 3: | 0— reserved for future use |
| Bit 4: | 0— reserved for future use |
| Bit 5: | 0— reserved for future use |

**Note 4**: Encrypted/Hash data sent status.

| Field 9: Encrypted data sent status | |
|---|---|
| **Bit 0:** | 1— track 1 encrypted data present |
| **Bit 1:** | 1— track 2 encrypted data present |
| **Bit 2:** | 1— track 3 encrypted data present |
| **Bit 3:** | 1— track 1 hash data present |
| **Bit 4:** | 1— track 2 hash data present |
| **Bit 5:** | 1— track 3 hash data present |
| **Bit 6:** | 1—session ID present |
| **Bit 7:** | 1—**KSN** present |

## 10.6. Additional Description

Except for USBKB and PS2 interfaces, track formatting (preamble, prefix, and separators) is not supported in a reader set to send encrypted track data.

The track data is always sent in the same format. There is no special formatting so the program doing the decoding knows where data field is located.

For USBKB and PS2 interfaces, Preamble and Postamble will be available in the encrypted track data

### 10.6.1. T1, T2 or T3 Data Length:

Each byte value indicates how many bytes of decoded card data are in the track data field. The value will be zero if there was no data on the track or if there was an error decoding the track.

The hashed data may be omitted while track 3 may be hashed and included.
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). The expiration date is masked by default but can be optionally displayed.

**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 and the key used for encrypting data is called the **Data Key**. **Data Key** is generated by first taking the **DUKPT Derived Key** exclusive or'ed with `0000000000FF0000 0000000000FF0000` to get the resulting intermediate variant key.

The left side of the intermediate variant key is then **TDES** encrypted with the entire 16-byte variant as the key. After the same steps are performed for the right side of the key, combine the two key parts to create the **Data Key**.

## 10.7. How to get Encrypted Data Length

The encrypted track data 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.

In the original format, 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, and add the Track 1, Track 2 and Track 3 together. Round up the total length by 8 if it's **TDES** or 16 for **AES**.

In enhanced format, the tracks data are encrypted separately rather than as a group.
To calculate the encrypted track length for each track, round up the track unencrypted data length by 8 for **TDES** or 16 for **AES**.

For example, to calculate the encrypted track 1 length, round up the track 1 unencrypted data length (field 5) by 8 for **TDES** or16 if it's **AES**.

Please refer to section 11.1 Decryption Samples for detailed samples.

**Track 1, 2, and 3 hashed**
SecureMag reader uses SHA-1 to generate hashed data for both track 1, track 2 and track 3 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 prevents unexpected noise in data transmission.
- 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.

Some Additional notes:

1. Track status byte is defined as the following:

| | **Track Status (bit 0, 1, 2: T 1 , 2, 3 decode; bit 3, 4, 5: T 1, 2, 3 sampling)** | | | | | |
|---|---|---|---|---|---|---|
| | **Sampling** | | | **Decoding** | | |
| | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| Track 1 Empty | | | 0 | | | 1 |
| Track 2 Empty | | 0 | | | 1 | |
| Track 3 Empty | 0 | | | 1 | | |
| | | | | | | |
| Track 1 Decode | | | 1 | | | 1 |
| Track 2 Decode | | 1 | | | 1 | |
| Track 3 Decode | 1 | | | 1 | | |
| | | | | | | |
| Track 1 Fail to Decode | | | 1 | | | 0 |
| Track 2 Fail to Decode | | 1 | | | 0 | |
| Track 3 Fail to Decode | 1 | | | 0 | | |

2. Please be aware that track status byte in secured output is different from track status bytes in OPOS head (called read error1 and read error2). OPOS header will only be used in OPOS mode security level 1 and level 2 and secure output only used in level 3 or level 4.
3. For **USB HID Secure Output,** the output format is same as **Secure Output** structure. No HID header is added. But the total length is the HIDSIZE (560 bytes as described in USB enumeration. HIDSIZE is subject to change. Software should auto adjust in case enumeration changes). Unused bytes will be filled with 0x00. This applied to secure Level 3 and Level 4 output, whether or not the data is encrypted.
4. Examples for field 8 (Clear/mask data sent status) and field 9 (Encrypted/Hash data sent status) These two bytes are omitted in original structure. In the enhanced encrypt structure, these two bytes are used to indicate the presence of each track's Clear or Masked data, Encrypted data and hash data.

   **Example**:
   field `8 = 0x03 (00000011)`
   field `9 = 0xBF (10111111)`
   **T1**: Mask data present; Encrypted data present; Hash present
   **T2**: Mask data present; Encrypted data present; Hash present
   **T3**: No Mask data; Encrypted data present; Hash present
   **KSN**: present
   **Session ID**: not present.

# 11. Additional Settings

**Send LRC in secured mode** (6F)

`53 6F 01 31` // to send LRC in secure mode (Default)

`53 6F 01 30` // Remove LRC in secure mode

**Display Expiration Data** (50)

`53 50 01 30` // Do not display Expiration Date (Exp date Masked) (Default)

`53 50 01 31` // Display Expiration Data

**Reader Serial Number** (4E)

The serial number will be set to the same as S/N in unit's label. The length is 8 to 10 characters. User can read out the S/N with `52 4E` command.

## 11.1. Decryption Example

Key for all examples:

```
0123456789ABCDEFFEDCBA9876543210
```

### 11.1.1. Security Level 3 Decryption - Original Encryption Format

Decryption of a three track ABA card with the original encryption format.

## 11.2. SecureMag Reader with Default Settings

Original encryption format can be recognized because the high bit of the fourth byte underlined (00) is 0.

```
027D01003F48236B252A343236362A2A2A2A2A2A2A2A393939395E42555348204A522F
47454F52474520572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
2A2A2A2A2A2A2A2A3F2A3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A
2A2A2A2A2A2A2A2A3F2A3B333333333333333333333736373637363037303730373736
3736373633333333333333333333373637363736303730373037373637363736333333
3333333333333337363736373630373037303737363736373633333333333333333333
373637363736303730373F32863E9E3DA28E455B28F7736B77E47A64EDDA3BF03A06E4
4F31D1818C0BCD7A353FB1AD70EFD30FFC3DA08A4FBC9372E57E8B40848BAEAA3FE724
B3550E2F4B223E6BF264BEAE9E39142B648CDB51FB8DAF8EA5B63913D29419B67582FC
CCE9B372660F03668CC453216D9449C6B67EF33418AC88F65E1DB7ED4D10973F99DFC8
463FF6DF113B6226C4898A9D355057ECAF11A5598F02CA316299490119000000000139
9F03
```

STX, Length (LSB, MSB), card type, track status, length track 1, length track 2, length track 3

```
02 7D01 00 3F 48 23 6B
```

The above broken down and interpreted

| | |
|---|---|
| **02** | —STX character |
| **7D** | —low byte of total length |
| **01** | —high byte of total length |
| **00** | —card type byte (interpretation old format ABA card) |
| **3F** | —3 tracks of data all good |
| **48** | —track 1 clear/mask data length |
| **23** | —track 2 clear/mask data length |
| **6B** | — track 3 clear/mask data length |

**Track 1 data masked (length 0x48)**

```
252A343236362A2A2A2A2A2A2A2A393939395E42555348204A522F47454F5247452057
2E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
3F2A
```

**Track 2 data in hex masked (length 0x23)**

```
3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3F2A
```

**Track 3 data unencrypted (length 0x6B)**

```
3B333333333333333333333736373637363037303730373736373637363333333333333
3333333337363736373630373037303737363736373633333333333333333333373637
3637363037303730373736373637363333333333333333333333373637363736303730 37
3F32
```

**Track 1 & 2 encrypted length 0x48+0x23 rounded up to 8 bytes =0x6B -> 0x70 (112 decimal)**

```
863E9E3DA28E455B28F7736B77E47A64EDDA3BF03A06E44F31D1818C0BCD7A353FB1AD
70EFD30FFC3DA08A4FBC9372E57E8B40848BAEAA3FE724B3550E2F4B
223E6BF264BEAE9E39142B648CDB51FB8DAF8EA5B63913D29419B67582FCCCE9B37266
0F03668CC453216D9449C6B67EF3
```

**Track 1 hashed**

```
3418AC88F65E1DB7ED4D10973F99DFC8463FF6DF
```

**Track 2 hashed**

```
113B6226C4898A9D355057ECAF11A5598F02CA31
```

**KSN**

```
62994901190000000001
```

**LRC, checksum and ETX**

```
39 9F 03
```

**Masked Data:**

**Track 1 data masked in ASCII**

```
%*4266********9999^BUSH JR/GEORGE
W.MR^*******************************?*
```

**Track 2 data masked in ASCII**

```
;4266********9999=***************?*
```

**Track 3 data unencrypted in ASCII**

```
;3333333333767676070707767676333333333376767607070776767633333333333767
67607070776767633333333337676760707?2
```

**Key Value**

```
F8 2A 7A 0D 7C 67 46 F1 96 18 9A FB 54 2C 65 A3
```

**KSN**

```
62 99 49 01 19 00 00 00 00 01
```

**Decrypted Data in ASCII**

```
%B4266841088889999^BUSH JR/GEORGE
W.MR^0809101100001100000000046000000?!;4266841088889999=08091011000004
6?0
;3333333333767676070707767676333333333376767607070776767633333333333767
67607070776767633333333337676760707?2
```

**Decrypted Data in Hex**

```
2542343236363834313038383838393939395E42555348204A522F47454F5247452057
2E4D525E30383039313031313030303031313030303030303030303436303030303030
3F213B34323636383431303838383839393939393D303830393130313130303030303436
3F300000000000
```

### 11.2.1. Security Level 4 Decryption - Original Encryption Format

```
028501003F48236B252A343236362A2A2A2A2A2A2A2A393939395E42555348204A522F
47454F52474520572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
2A2A2A2A2A2A2A2A3F2A3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A
2A2A2A2A2A2A2A2A3F2A3B33333333333333333333373637363736303730373037373736
373637363333333333333333333333373637363736303730373037373637363736333333
3333333333333333373637363736303730373037373637363736333333333333333333
3736373637363037303733F32ED9DB728814F150D177F769B0441C52B2B1994C83D058F
1DDA5DAA6753CF0F61BB7690C7E8A276D3D606513D1F8B79423C70594A0849CBB4C7B5
A8DAC2B1A21B11F1C47EF4F12AC07D59A79E9369372D3F906A7F6C6D2B9076BCF05B33
4441FAEC8B4EFBEB9DD20EBF97B29D910C415FCEA8DA8FEB9775343418AC88F65E1DB7
ED4D10973F99DFC8463FF6DF113B6226C4898A9D355057ECAF11A5598F02CA31629949
011900000000044B6F03
```

**Masked Data**

**Track 1**

```
%*4266********9999^BUSH JR/GEORGE
W.MR^*******************************?*
```

**Track 2**

```
;4266********9999=***************?*
```

**Track 3**

```
;3333333333767676070707767676333333333376767607070776767633333333333767
6760707077676763333333333767676 0707?2
```

**Key Value**

```
8A 92 F6 74 00 BF 25 2E 57 9A A9 01 FF 27 48 41
```

**KSN**

```
62 99 49 01 19 00 00 00 00 04
```

**Session ID**

AA AA AA AA AA AA AA AA

**Decrypted Data in ASCII**

```
%B4266841088889999^BUSH JR/GEORGE
W.MR^0809101100001100000000046000000?!;4266841088889999=08091011000004
6?0
;3333333333767676070707767676333333333376767607070776767633333333333767
67607070776767633333333337676760707?2
```

**Decrypted Data in Hex**

```
2542343236363834313038383838393939395E42555348204A522F47454F5247452057
2E4D525E30383039313031313030303031313030303030303030303436303030303030
3F213B343236363834313038383838393939393D303830393130313130303030303436
3F30AAAAAAAAAAAAAAAA0000000000
```

### 11.2.2. Security Level 3 Decryption - Enhanced Encryption Format

This is an example of enhanced encryption format (this can be recognized because the high bit of the fourth byte underlined (80) is 1.

```
029801803F48236B03BF252A343236362A2A2A2A2A2A2A2A393939395E42555348204A
522F47454F52474520572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
2A2A2A2A2A2A2A2A2A2A3F2A3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A
2A2A2A2A2A2A2A2A2A2A3F2ADA7F2A52BD3F6DD8B96C50FC39C7E6AF22F06ED1F033BE
0FB23D6BD33DC5A1F808512F7AE18D47A60CC3F4559B1B093563BE7E07459072ABF8FA
AB5338C6CC8815FF87797AE3A7BEAB3B10A3FBC230FBFB941FAC9E82649981AE79F263
2156E775A06AEDAFAF6F0A184318C5209E55AD44A9CCF6A78AC240F791B63284E15B40
```

```
19102BA6C505814B585816CA3C2D2F42A99B1B9773EF1B116E005B7CD8681860D174E6
AD316A0ECDBC687115FC89360AEE7E430140A7B791589CCAADB6D6872B78433C3A25DA
9DDAE83F12FEFAB530CE405B701131D2FBAAD970248A456000933418AC88F65E1DB7ED
4D10973F99DFC8463FF6DF113B6226C4898A9D355057ECAF11A5598F02CA31688861C1
57C1CE2E0F72CE0F3BB598A614EAABB16299490119000000000206E203
```

STX, Length (LSB, MSB), card type, track status, length track 1, length track 2, length track 3

```
02 9801 80 3F 48-23-6B 03BF
```

The above broken down and interpreted:

| | |
|---:|---|
| **02** | —STX character |
| **98** | —low byte of total length |
| **01** | —high byte of total length |
| **80** | —card type byte (interpretation new format ABA card) |
| **3F** | —3 tracks of data all good |
| **48** | —length of track 1 |
| **23** | —length of track 2 |
| **6B** | —length of track 3 |
| **03** | —tracks 1 and 2 have masked/clear data |
| **BF** | —bit 7=1—**KSN** included |
| **Bit 6=0** | —no Session ID included so not level 4 encryption |
| **Bit 5=1** | —track 3 hash data present |
| **Bit 4=1** | —track 2 hash data present |
| **Bit 3-1** | —track 1 hash data present |
| **Bit 2=1** | —track 3 encrypted data present |
| **Bit 1=1** | —track 2 encrypted data present |
| **Bit 0=1** | —track 1 encrypted data present |

**Track 1 data masked (length 0x48)**

```
252A343236362A2A2A2A2A2A2A2A393939395E42555348204A522F47454F5247452057
2E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
3F2A
```

**Track 1 masked data in ASCII**

```
%*4266********9999^BUSH JR/GEORGE
W.MR^*******************************?*
```

**Track 2 data in hex masked (length 0x23)**

```
3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3F2A
```

**Track2 masked data in ASCII**

```
;4266********9999=***************?*
```

In this example there is no Track 3 data either clear or masked (encrypted and hashed data is below)

**Track 1 encrypted length 0x48 rounded up to 8 bytes = 0x48 (72 decimal)**

```
DA7F2A52BD3F6DD8B96C50FC39C7E6AF22F06ED1F033BE0FB23D6BD33DC5A1F808512F
7AE18D47A60CC3F4559B1B093563BE7E07459072ABF8FAAB5338C6CC8815FF87797AE3
A7BE
```

**Track 2 encrypted length 0x23 rounded up to 8 bytes =0x28 (40 decimal)**

```
AB3B10A3FBC230FBFB941FAC9E82649981AE79F2632156E775A06AEDAFAF6F0A184318
C5209E55AD
```

**Track 3 encrypted length 0x6B rounded up to 8 bytes =0x70 (112 decimal)**

```
44A9CCF6A78AC240F791B63284E15B4019102BA6C505814B585816CA3C2D2F42
A99B1B9773EF1B116E005B7CD8681860D174E6AD316A0ECDBC687115FC89360A
EE7E430140A7B791589CCAADB6D6872B78433C3A25DA9DDAE83F12FEFAB530CE405B70
1131D2FBAAD970248A45600093
```

**Track 1 data hashed length 20 bytes**

```
3418AC88F65E1DB7ED4D10973F99DFC8463FF6DF
```

**Track 2 data hashed length 20 bytes**

```
113B6226C4898A9D355057ECAF11A5598F02CA31
```

**Track 3 data hashed length 20 bytes**

```
688861C157C1CE2E0F72CE0F3BB598A614EAABB1
```

**KSN length 10 bytes**

```
62994901190000000002
```

**LCR, check sum and ETX**

```
06E203
```

**Clear/Masked Data in ASCII**

**Track 1**

```
%*4266********9999^BUSH JR/GEORGE
W.MR^*******************************?*
```

**Track 2**

```
;4266********9999=***************?*
```

**Key Value**

```
1A 99 4C 3E 09 D9 AC EF 3E A9 BD 43 81 EF A3 34
```

**KSN**

```
62 99 49 01 19 00 00 00 00 02
```

## 11.3. Decrypted Data

**Track 1 decrypted**

```
%B4266841088889999^BUSH JR/GEORGE
W.MR^0809101100001100000000046000000?!
```

**Track 2 decrypted**

```
;4266841088889999=080910110000046?0
```

**Track 3 decrypted**

```
;3333333333767676070707767676333333333376767607070776767633333333333767
676070707767676333333333376767607 07?2
```

**Track 1 decrypted data in hex including padding zeros (but there are no pad bytes here)**

```
254234323636383431303838383839393939 5E42555348204A522F47454F5247452057
2E4D525E30383039313031313030303031313030303030303030303034363030303030 30
3F21
```

**Track 2 decrypted data in hex including padding zeros**

```
3B343236363834313038383838393939393D3038303931303131303030303034363F30
0000000000
```

**Track 3 decrypted data in hex including padding zeros**

```
3B333333333333333333333736373637363037303730373736373637363333333333333
3333333337363736373630373037303737363736373633333333333333333333333373637
3637363037303730373736373637363333333333333333333333373637363736303730 37
3F320000000000
```

**Enhanced Encryption Format**

```
02A001803F48236B03FF252A343236362A2A2A2A2A2A2A2A393939395E42555348204A
522F47454F52474520572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
2A2A2A2A2A2A2A2A2A2A3F2A3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A
2A2A2A2A2A2A2A2A2A2A3F2A6D7D5B204D3579694E148F3FB2565544D35825EA89BA30
C966D34363151BF592F995EDA86B94A47EBFDF6434CB3A075DDD18F616E21F1E2038BC
3AD5F96C1387177BD89409DA2E92A684543E007087F8694AEA8D3DB36BA10BC4D4B277
1C622FEC8271A6E021AA564
```

```
4ED559EC09CABF19F36B422CA2016B48A7241B2DA9584ED4415B4F30637734CF5031AF
475DAF27C188A1A771264011BAA090E91893BC2A52EDD56F8E6E9554BC0C5207C04E3C
21B6DA2A48F2257DC6946DBFBC87F3189E5C8B954BF7303D01E443155911E4137AEAD5
2441567AA1D50924A7597EC9D758AB4F3A8E82BF81A2E3418AC88F65E1DB7ED4D10973
F99DFC8463FF6DF113B6226C4898A9D355057ECAF11A5598F02CA31688861C157C1CE2
E0F72CE0F3BB598A614EAABB162994901190000000003D67C03
```

**Clear/Masked Data**

**Track 1**

```
%*4266*******9999^BUSH JR/GEORGE
W.MR^*******************************?*
```

**Track 2**

```
;4266*******9999=**************?*
```

**Key Value**

```
89 52 50 33 61 75 51 5C 41 20 CF 45 F4 1A BF 1C
```

**KSN**

```
62 99 49 01 19 00 00 00 00 03
```

**Session ID**

```
AA AA AA AA AA AA AA AA
```

**Decrypted Data in ASCII**

```
%B4266841088889999^BUSH JR/GEORGE
W.MR^0809101100001100000000046000000?!
;4266841088889999=0809101100000046?0
;3333333333767676070707767676333333333376767607070776767633333333333767
67607070776767633333333337676760707?2
```

**Decrypted Data in Hex**

```
2542343236363834313038383838393939395E42555348204A522F47454F5247452057
2E4D525E30383039313031313030303031313030303030303030303436303030303030
3F213B343236363834313038383838393939393D30383039313031313030303030303436
3F3000000000003B333333333333333333333736373637363037303730373736373637
36333333333333333333333373637363736303730373037373637363736333333333333
333333333736373637363037303730373736373637363333333333333333333333373637
363736303730373F320000000000
```

### 11.3.1. Level 4 Activate Authentication Sequence

The security level changes from 3 to 4 when the device enters authentication mode successfully. Once the security level is changed to level 3 or 4, it cannot go back to a lower level.

### 11.3.2. Activate Authentication Mode Command

When the reader is in security level 4, it would only transmit the card data when it is in **Authenticated Mode**.

## 11.4. Authentication Mode Request

When sending the **Authentication Request**, the user also needs to specify a time limit for the reader to wait for the **Activation Challenge Reply** command.

The minimum timeout duration required is 120 seconds. If the specified time is less than the minimum, 120 seconds would be used for timeout duration. The maximum time allowed is 3600 seconds (one hour). If the reader times out while waiting for the **Activation Challenge Reply**, the authentication failed.

## 11.5. Device Response

When **Authentication Mode** is requested, the device responds with two challenges: Challenge 1 and challenge 2. The challenges are encrypted using the current **DUKPT Key** exclusive- or'ed with `<F0F0 F0F0 F0F0 F0F0 F0F0 F0F0 F0F0 F0F0>`.

The decrypted challenge 1 contains 6-bytes of random number followed by the last 2-bytes of **KSN**. The 2-bytes of **KSN** may be compared with the last 2-bytes of the clear text **KSN** sent in the message to authenticate the reader. The user should complete the **Activate Authentication** sequence using **Activation Challenge Reply** command.

**Command Structure**

Host -> Device:
`<STX><R><80h><02h><Pre-Authentication Time Limit><ETX><LRC>`

Device -> Host:
`<ACK><STX><Device Response Data><ETX><LRC>` (success)
`<NAK>` (fail)

### 11.5.1. Pre-Authentication Time Limit

2-bytes of time in seconds
**Device Response Data**: 26-bytes data, consists of `<Current Key Serial Number>`
`<Challenge 1>`

`<Challenge 2>`
**Current Key Serial Number**: 10-bytes data with **Initial Key Serial Number** in the leftmost 59 bits and **Encryption Counter** in the rightmost 21 bits.

1. **Challenge 1**: 8-bytes challenge used to activate authentication. Encrypted using the key derived from the current **DUKPT Key**.
2. **Challenge 2**: 8-bytes challenge used to deactivate authentication. Encrypted using the key derived from the current **DUKPT Key**.

## 11.6. Activation Challenge Reply Command

This command serves as the second part of an **Activate Authentication** sequence. The host sends the first 6-bytes of Challenge 1 from the response of **Activate Authenticated Mode** command, 2-bytes of

**Authenticated Mode** timeout duration, and 8-bytes Session ID encrypted with the result of current **DUKPT Key** exclusive- or'ed with `<3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C>`.

The **Authenticated Mode** timeout specifies the maximum time (in seconds) in which a reader would remain in **Authenticated Mode**. A value of zero forces the reader to stay in **Authenticated Mode** until a card swipe or power down occurs. The minimum timeout duration required is 120 seconds. If the specified time is less than the minimum, 120 seconds would be used for timeout duration.

If Session ID information is included and the command is successful, the Session ID will be changed.

The **Activate Authenticated Mode** succeeds if the device decrypts **Challenge Reply** responds correctly. If the device cannot decrypt **Challenge Reply** command, **Activate Authenticated Mode** fails and **DUKPT KSN** advances.

**Command Structure**

Host -> Device:
`<STX><S><82h><08h><Activation Data><ETX><LRC>`

Device -> Host:
`<ACK>` (success)
`<NAK>` (fail)

### 11.6.1. Activation Data

8 or 16-bytes, structured as `<Challenge 1 Response> <Session ID>`

**Challenge 1 Response**: 6-bytes of Challenge 1 random data with 2-bytes of **Authenticated Mode** timeout duration. It's encrypted using the key derived from the current **DUKPT Key**.

**Session ID**: Optional 8-bytes Session ID, encrypted using the key derived from the current **DUKPT Key**.

## 11.7. Deactivate Authenticated Mode Command

This command is used to exit **Authenticated Mode**. The Host needs to send the first 7-bytes of Challenge 2 (from the response of **Activate Authenticated Mode** command) and the **Increment Flag** (0x00 indicates no increment, 0x01 indicates increment of the **KSN**) encrypted with current **DUKPT Key** exclusive- or'ed with `<3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C>`.

If device decrypts Challenge 2 successfully, the device will exit **Authenticated Mode**. The **KSN** will increase if the **Increment Flag** is set to 0x01. If device cannot decrypt Challenge 2 successfully, it will stay in **Authenticated Mode** until a timeout occurs or when customer swipes a card.

The **KSN** is incremented every time the **Authenticated Mode** is exited by a timeout or card swipe. When the **Authenticated Mode** is exited by the **Deactivate Authenticated Mode** command, the **KSN** will increase when the increment flag is set to 0x01.

**Command Structure**

Host -> Device:
`<STX><S><83h><08h><Deactivation Data><ETX><LRC>`

Device -> Host:
`<ACK>` (success)
`<NAK>` (fail)

`<Deactivation Data>`: 8-bytes response to Challenge 2. It contains 7-bytes of Challenge 2 with 1-byte of **Increment Flag**, encrypted by the specified variant of current **DUKPT Key**

## 11.8. Get Reader Status Command

**Command Structure**

Host -> Device:
`<STX><R><83h><ETX><LRC>`

Device -> Host:
`<ACK><STX><83h><02h><Current Reader Status><Pre-Condition><ETX><LRC>`
(success)
`<NAK>`  (fail)

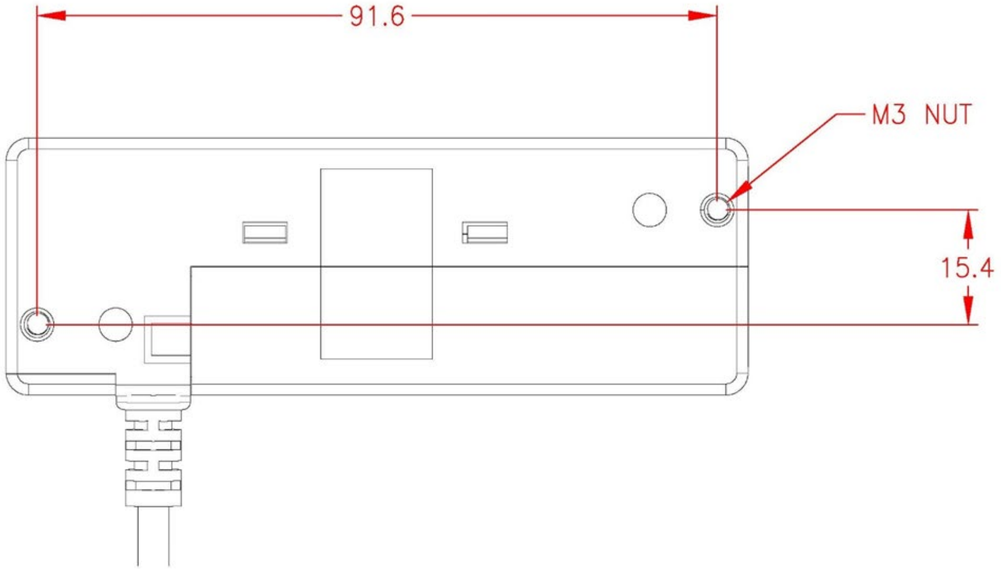
**Current Reader Status**: 2-bytes data with1-byte of `<Reader State>` and 1-byte of `<Pre-Condition>`

| **Reader State:** | Indicates the current state of the reader. |
|---|---|
| **0x00:.** | The reader is waiting for **Activate Authentication Mode Command**. The command must be sent before the card can be read. |
| **0x01:** | The authentication request is sent, and the reader is waiting for the **Activation Challenge Reply Command**. |
| **0x02:** | The reader is waiting for a card swipe. |

| **Pre-condition:** | Specifies how the reader goes to its current state as follows |
|---|---|
| **0x00:** | The reader has no card swipes and has not been authenticated since it was powered up. |
| **0x01:** | **Authentication Mode** was activated successfully. The reader processed a valid **Activation Challenge Reply** command |
| **0x02:** | The reader receives a good card swipe. |
| **0x03:** | The reader receives a bad card swipe, or the card is invalid. |
| **0x04:** | **Authentication Activation Failed**. |
| **0x05:** | **Authentication Deactivation Failed**. |
| **0x06:** | **Authentication Activation Timed Out**: The host fails to send an **Activation Challenge Reply** command within the time specified in the **Activate Authentication Mode** command. |
| **0x07:** | **Swipe Timed Out**: The user fails to swipe a card within the time specified in the **Activation Challenge Reply** command. |

# 12. Appendix A: Mounting Dimension Information

The bottom of the reader must be flat for mounting. If the reader needs to be mounted on the table, please unscrew the two screws showed in red below. Those screws allow the two holes for mounting. The mounting nut is M3x 3.



# 13. Appendix B: Setting Configuration Parameters and Values

Not all Function IDs are present in different hardware versions of the SecureMag.

The codes in this list reflect the last row in the table below:

- The '-' feature is not currently supported and exists for compatibility.
- The 's' feature is available in the RS232 serial version of the reader.
- The 'u' feature is available only in the USB version.
- The 'k' feature is available in the keyboard version.
- The 'p' feature is available only in the SPI version.
- The 'r' "Reset All" does not affect this value.
- The 'n' is not directly settable.
- The 'd' feature is only for reader with the data editing feature.
- The 'e' feature is only for reader with encrypt feature.
- The 'i' feature is ignored for encrypted transactions.

Most Function ID settings related to formatting the track output do not work in **Secure Mode**. Exceptions to this are Preamble and Postamble in keyboard mode only. It is currently not possible to mix security with OPOS and JPOS support.

# 14. Appendix B: Setting Configuration Parameters and Values

The following is a table of default setting and available settings (value within parentheses) for each Function ID.

| Function ID | Hex | Description | Default Setting | Description | |
|---|---|---|---|---|---|
| HTypeID | 10 | Terminal Type | '0' (‘0’~’2’,'4'~'6') | PC/AT, Scan Code Set 2, 1, 3, PC/AT with external Keyboard and PC/AT without External Keyboard | kr |
| BeepID | 11 | Beep Setting Beep Frequency And Duration | ‘2’ (‘0’~’4’) | ‘0’ no beep; ‘1’ low long; ‘2’ high long; ‘3’ high short; low short; | |
| ChaDelayID | 12 | Character Delay | ‘0’ (‘0’~’5’)<br>‘6’ | 2 ms inter-character delay<br>‘6 for 0 mS delay | k |
| TrackSelectID | 13 | Track Selection | ‘0’ (‘0’~’9’)<br>**0x30 – Any Track**<br>0x31 – Track 1 Only<br>0x32 – Track 2 Only<br>0x33 – Track 1 & Track 2<br>0x34 – Track 3 Only<br>0x35 – Track 1 & Track 3<br>0x36 – Track 2 & Track 3<br>0x37 – All Three Tracks<br>0x38 – Track 1 Or Track 2<br>0x39 – Track 2 Or Track 3 | Any Track 0-any; 1-7—bit 1 tk1, bit 2 tk2; bit 3 tk3. ‘8’—tk1-2; ‘9’ tk2-3<br>If a track is not selected in a secure reader that track is not processed or recognized. | |
| PollingInterval ID | 14 | Polling Interval | 1 (1 ~ 255) | USB HID Polling Interval | u |
| DataFmtID | 15 | Data Output Format | ‘0’ (‘0’~’2’) | 0’-ID TECH Format; ‘1’-UIC; ‘2’-Mag-Tek Format | |
| FmtOptionID | 16 | UIC, Mag-Tek | H’59’ | Refer to MiniMag RS232 User’s Manual | |
| TrackSepID | 17 | Track Separator | 0x0D=CR/Enter | CR for RS232, Enter for KB any character supported except 00 which means none. | |
| SendOptionID | 19 | Send Option | ‘1’ (‘0’~0x3F) | Sentinel and Account Number Control<br>**0x30** -Does not send start/end sentinel and sends all data on Track 2, not an error notification. **Control Key Output**.<br>**0x31** - Sends start/end sentinel and all data on Track 2, does not send an error notification. **Control Key Output**.<br>**0x32** – Does not send start/end sentinel and only sends account number on Track 2. Does not | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | send error notification. **Control Key Output**.<br>**0x33** - Sends start/end sentinel and only sends account number on Track 2, does not send error notification. **Control Key Output**.<br>**0x34 - Not send start/end sentinel and send all data on Track 2, send error notification(default). Control Key Output**.<br>0x35 - Sends start/end sentinel, all data on Track 2, and error notification. **Control Key Output**.<br>**0x36** – Does not send start/end sentinel and only sends account number on Track 2. Also sends error notification. **Control Key Output**.<br>**0x37** - Send start/end sentinel and only send account number on Track 2, send error notification. **Control Key Output**.<br>**0x38** – Does not send start/end sentinel and sends all data on Track 2, no error notification. **Alt Key Output**.<br>**0x39** - Sends start/end sentinel and sends all data on Track 2, not sends error notification. **Alt Key Output**.<br>**0x3a** – Does not send start/end sentinel, only sends account number on Track 2, and does not send error notification**. Alt Key Output**.<br>**0x3b** - Send start/end sentinel and only send account number on Track 2, not send error notification. **Alt Key Output**.<br>**0x3c** - Not send start/end sentinel and send all data on Track 2, send error notification(default). **Alt Key Output**.<br>**0x3d** - Sends start/end sentinel, sends all data on Track | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 2, and sends error notification**.**<br>**Alt Key Output**.<br>**0x3e** – Does not send start/end sentinel, only sends account number on Track 2, and sends error notification. **Alt Key Output**.<br>**0x3f** - Sends start/end sentinel, only sends account number on Track 2, and sends error notification. **Alt Key Output**. | |
| MSRReadingID | 1A | MSR Reading | `'1' ('0'~'3')` | Enable/Disable MSR Reading<br>0x30 – MSR Reading Disabled<br>**0x31 – MSR Reading Auto Mode Enabled** 0x32 – MSR Reading Buffered Mode Enabled<br>0x33 Auto MSR Buffered Mode Enabled | |
| DTEnableSendID | 1B | DT Enable Send | `'0'('0','1','3')` | Data Editing Control<br>**0x30 – Disable Data Edit.**<br>**0x31** – Data Edit Match mode.<br>**0x33** – Data Edit Un-match mode | id |
| DecodingMethodID | 1D | Decoding Direction | `'1' ('0'~'3')` | Reading Direction<br>**0x30** – Raw Data Decoding in Both Directions.<br>**0x31 – Decode in Both directions.**<br>**0x32** – Moving Stripe Along Head in Direction of Encoding.<br>**0x33** – Moving Stripe Along Head Against Direction of Encoding. | |
| ReviewID | 1F | Review All Settings | `None` | | |
| TerminatorID | 21 | Terminator | `0x0D (any)` | CR for RS232, Enter for KB | i |
| FmVerID | 22 | Firmware Version | `None` | | |
| USBHIDFmtID | 23 | USB HID Fmt (HID rdr only) | `'0' ('0'~'1', '8')` | '0' ID TECH Format; '1' Mag-Tek Format; '8' HIDKB format | ur |
| ForeignKBID | 24 | Foreign KB | `'0' ('0' ~0x3A)` | Foreign Keyboard | k |
| CustSetID | 30 | Custom Customer Settings | `00(00-07)` | **.0 POS-X**: Level 3 Non-CC send same as Level1<br>**.1 Level 3**: No empty pkt when not enough sampling bits .2 Enhanced Secured Output will have SN after hash | |
| Track1PrefixID | 34 | Track 1 Prefix | `0 (any string)` | No prefix for track 1, 6-character max | i |

| | | | | | |
|---|---|---|---|---|---|
| Track2PrefixID | 35 | Track 2 Prefix | 0 (any string) | No prefix for track 2, 6-character max | i |
| Track3PrefixID | 36 | Track 3 Prefix | 0 (any string) | No prefix for track 3, 6-character max | i |
| Track1SuffixID | 37 | Track 1 Suffix | 0 (any string) | No suffix for track 1, 6-character max | i |
| Track2SuffixID | 38 | Track 2 Suffix | 0 (any string) | No prefix for track 2, 6-character max | i |
| Track3SuffixID | 39 | Track 3 Suffix | 0 (any string) | No prefix for track 3, 6-character max | i |
| PinKeyID | 3E | | 0x00,0x5A | 0x00-Data Key; 0x5A– PinKey<br>Can only set at secure level 1; | r |
| BaudID | 41 | Baud Rate | ‘5’ (‘2’~’9’) | 9600 bps, ‘2’ is 1200, ‘7’ is 38,400 bps; ‘9’ is 115.2 kbps | s |
| DataID | 42 | Data Pit | ‘0’ (‘0’~’1’) | ‘0’-8 Bits required in secure mode’1’-7 bits | s |
| ParityID | 43 | Data Parity | ‘0’ (‘0’~’4’) | ‘0’-None ‘1’- Even, ‘2’-Odd, ‘3’-Mark or ‘4’-Space | s |
| HandID | 44 | Handshake | ‘0’ (‘0’~’1’) | Software (Xon/Xoff) handshake | s |
| StopID | 45 | Stop Bit | ‘0’ (‘0’~’1’) | ‘0’-1 stop Bit; ‘1’-2 stop bits | s |
| XOnID | 47 | XOn Character | DC1 | 0x11 as XOn | s |
| XOffID | 48 | XOff Character | DC3 | 0x13 as XOff | s |
| PrePANID | 49 | PAN to not mask | 4 (0-6) | # leading PAN digits to display | e |
| PostPANID | 4A | PAN to not mask | 4 (0-4) | # of trailing PAN digits to display | e |
| MaskCharID | 4B | mask the PAN with this character | ‘*’ 20-7E | Any printable character | e |
| CrypTypeID | 4C | encryption type | ‘0’ (‘0’-‘2’) | ‘0’ no encryption ‘1’ 3DES ‘2’ AES | r<br>e |
| OutputModeID | 4D | Std, OPOS or JPOS | '0' ('0' ~ '2')<br>Reader does not save in non-volatile memory. | ‘0’-Standard mode; ‘1’ OPOS; ‘2’-JPOS<br>Always returns to ‘0’ on power-on. | |
| SerialNumberID | 4E | device serial # | any 8-10 bytes | 8-10-digit serial number; Can be set only once | r |
| DispExpDateID | 50 | mask or display expiration date | ‘0”0’-‘1’ | ‘0’ mask expiration date; ‘1’ display expiration date | e |
| SessionID | 54 | 8-byte hex not stored in EEPROM | None | always init to all ‘FF’ | e |
| Mod10ID | 55 | include mod10 check digit | '0' ('0'-'2') | ‘0’ don’t include mod10, ‘1’ display mod10, ‘2’ display wrong mod10 | e |
| DesKeyID | 56 | DES Key Value | 0 | internal use only | r<br>e |
| AesKeyID | 57 | AES Key Value | 0 | internal use only | r<br>e |
| KeyManageTypeID | 58 | DUKPT | ‘1’(‘0’-‘1’) | ‘0’ fixed key ‘1’ DUKPT | r |
| HashOptID | 5C | '7' (‘0’-‘7’) | Send tk1-2 hash bit 0:1<br>send tk1 hash; bit 1:1 | | e |

| | | | | | |
|---|---|---|---|---|---|
| | | | send tk2 hash; bit2:1<br>send tk3 hash. | | |
| HexCaseID, | | '1' (‘0’-‘1’) | ‘0’ send in lower case;<br>‘1’ send in upper case | | k |
| LRCID | 60 | track LRC | ‘0’ (‘0’~’1’) | ‘0’ send without track LRC in output; ‘1’ with track LRC | i |
| T17BStartID | 61 | Track 1 7 Bit Start Char | ‘%’ (any) | ‘%’ as Track 1 7 Bit Start Sentinel | i |
| T16BStartID | 62 | T16B Start | ‘%’ (any) | ‘%’ as Track 1 6 Bit Start Sentinel | i |
| T15BStartID | 63 | T15B Start | ‘;’ (any) | ‘;’ as Track 1 5 Bit Start Sentinel | i |
| T27BStartID | 64 | Track 2 7 Bit Start Char | ‘%’ (any) | ‘%’ as Track 2 7 Bit Start Sentinel | i |
| T25BStartID | 65 | T25BStart | ‘;’ (any) | ‘;’ as Track 2 5 Bit Start Sentinel | i |
| T37BStartID | 66 | Track 3 7 Bit Start Char | ‘%’ (any) | ‘%’ as Track 3 7 Bit Start Sentinel | i |
| T36BStartID | 67 | T36BStart | ‘!’ (any) | ‘!’ as Track 3 6 Bit Start Sentinel | i |
| T35BStartID | 68 | T35BStart | ‘;’ (any) | ‘;’ as Track 3 5 Bit Start Sentinel | i |
| T1EndID | 69 | Track 1 End Sentinel | ‘?’ (any) | ‘?’ as End Sentinel | i |
| T2EndID | 6A | Track 2 End Sentinel | '?' (any) | ‘?’ as End Sentinel | i |
| T3EndID | 6B | Track 3 End Sentinel | '?' (any) | ‘?’ as End Sentinel | i |
| T1ERRSTARTID | 6C | Track 1 error code | ‘%’ (any) | start sentinel if track 1 error report | i |
| T2ERRSTARTID | 6D | Track 2 error code | ‘;’ (any) | start sentinel if track 2 error report | i |
| T3ERRSTARTID | 6E | Track 3 error code | ‘+’ (any) | start sentinel if track 3 error report | i |
| SecureLrcID | 6F | Secured output format track Lrc option | ‘1’ (‘0’-‘1’) | ‘1’ to send track LRC in secured output data; ‘0’<br>don’t send track LRC | e |
| T28BStartID | 72 | JIS T12 SS/ES | 0x00 or 0x7F | 0 unless keyboard version then 0x7F | i |
| T38BStartID | 73 | JIS T3 SS/ES | 0x00 or 0x7F | 0 unless keyboard version then 0x7F | i |
| SPISettingID | 75 | | ‘0’ | | p |
| EquipFwID | 77 | feature option setting | any | Factory Reader firmware configuration setting | rn |
| SyncCheckID | 7B | check for track sync bits-can allow poorly encoded cards to be read | ‘0’ (‘0’-2’) | check leading & trailing sync bits ‘0’ 13 bits; ‘1’ 13 bits, but allow if valid through track LRC; '2' 9 bits ABA; 13 bits IATA; 16 bits JIS | |
| MagTSecureLvlID | 7D | | ‘1’ (‘0’-‘3’) | | p |
| SecurityLevelID | | Reader’s encryption level | ‘1’ or ‘3’ (‘0’-‘4’) | 1’ no encryption; ‘2’ key loaded; ‘3 encrypted reader; ‘0’ DUKPT exhausted; ‘4’ authentication required | nr |
| MagTCryptID | 7F | | '1'(‘0’-‘3’) | | p |

| | | | | | |
|---|---|---|---|---|---|
| EncryptOptID | | encryption options, enhanced only | `8 encrypt trk 3 if card type 0; (0-F)` | | |
| EncryptStrID | `85` | encrypt structure | `'0'` | '0' original; '1' enhanced | |
| MaskOptID | `86` | clear / mask data options | `7` | bit 0 send clear/mask trk1 bit 1 send clear/mask trk2 bit 2 send clear/mask trk3 | |
| PrefixID | `D2` | Preamble | `0 (any 15)` | No Preamble, 15 char max | 1 |
| PostfixID | `D3` | Postamble | `0 (any 15)` | No Postamble, 15 char max | |
| AddedFieldID | `FA` | Data Edit Added field | `0` | See Data Edit documentation. | i |
| SearchCmdID | `FB` | Data Edit Search cmd | `0` | See Data Edit documentation. | i |
| SendCmdID | `FC` | Data Edit send cmd | `0` | See Data Edit documentation. | i |
| SendCmd2ID | `FD` | Data Edit send cmd 2 | `0` | See Data Edit documentation. | i |

# 15. Appendix C: Key Code Table in USB Keyboard Interface

Check if “Cap Locks” is on before sending out code because most characters will be in reverse if it is on.

For Function code B1 to BA set "Num Lock", send out the code, and then clear it.

For Function code BB to C2, C9 to CC, if "Num Lock" is set then clear it and set it after finishing sending out code.

| Keystroke | Hex Value | Functional Code | USB KB Code |
|---:|---|---|---|
| Ctrl+2 | 00 | | 1F Ctrl On |
| Ctrl+A | 01 | | 04 Ctrl On |
| Ctrl+B | 02 | | 05 Ctrl On |
| Ctrl+C | 03 | | 06 Ctrl On |
| Ctrl+D | 04 | | 07 Ctrl On |
| Ctrl+E | 05 | | 08 Ctrl On |
| Ctrl+F | 06 | | 09 Ctrl On |
| Ctrl+G | 07 | | 0A Ctrl On |
| BS | 08 | \bs | 2A |
| Tab | 09 | \tab | 2B |
| Ctrl+J | 0A | | 0D Ctrl On |
| Ctrl+K | 0B | | 0E Ctrl On |
| Ctrl+L | 0C | | 0F Ctrl On |
| Enter | 0D | \enter | 28 |
| Ctrl+N | 0E | | 11 Ctrl On |
| Ctrl+O | 0F | | 12 Ctrl On |
| Ctrl+P | 10 | | 13 Ctrl On |
| Ctrl+Q | 11 | | 14 Ctrl On |
| Ctrl+R | 12 | | 15 Ctrl On |
| Ctrl+S | 13 | | 16 Ctrl On |
| Ctrl+T | 14 | | 17 Ctrl On |
| Ctrl+U | 15 | | 18 Ctrl On |
| Ctrl+V | 16 | | 19 Ctrl On |
| Ctrl+W | 17 | | 1A Ctrl On |
| Ctrl+X | 18 | | 1B Ctrl On |
| Ctrl+Y | 19 | | 1C Ctrl On |
| Ctrl+Z | 1A | | 1D Ctrl On |
| ESC | 1B | \esc | 29 |
| Ctrl+\ | 1C | | 31 Ctrl On |
| Ctrl+] | 1D | | 30 Ctrl On |
| Ctrl+6 | 1E | | 23 Ctrl On |
| Ctrl+- | 1F | | 2D Ctrl On |
| SPACE | 20 | | 2C |
| ! | 21 | | 1E Shift On |
| " | 22 | | 34 Shift On |
| # | 23 | | 20 Shift On |
| $ | 24 | | 21 Shift On |

| | | | |
|---|---|---|---|
| % | 25 | | 22 Shift On |
| & | 26 | | 24 Shift On |
| ' | 27 | | 34 |
| ( | 28 | | 26 Shift On |
| ) | 29 | | 27 Shift On |
| * | 2A | | 25 Shift On |
| + | 2B | | 2E Shift On |
| , | 2C | | 36 |
| - | 2D | | 2D |
| . | 2E | | 37 |
| / | 2F | | 38 |
| 0 | 30 | | 27 Shift On |
| 1 | 31 | | 1E Shift On |
| 2 | 32 | | 1F Shift On |
| 3 | 33 | | 20 Shift On |
| 4 | 34 | | 21 Shift On |
| 5 | 35 | | 22 Shift On |
| 6 | 36 | | 23 Shift On |
| 7 | 37 | | 24 Shift On |
| 8 | 38 | | 25 Shift On |
| 9 | 39 | | 26 Shift On |
| : | 3A | | 33 Shift On |
| ; | 3B | | 33 |
| < | 3C | | 36 Shift On |
| = | 3D | | 2E |
| > | 3E | | 37 Shift On |
| ? | 3F | | 38 Shift On |
| @ | 40 | | 1F |
| A | 41 | | 04 Shift On |
| B | 42 | | 05 Shift On |
| C | 43 | | 06 Shift On |
| D | 44 | | 07 Shift On |
| E | 45 | | 08 Shift On |
| F | 46 | | 09 Shift On |
| G | 47 | | 0A Shift On |
| H | 48 | | 0B Shift On |
| I | 49 | | 0C Shift On |
| J | 4A | | 0D Shift On |
| K | 4B | | 0E Shift On |
| L | 4C | | 0F Shift On |
| M | 4D | | 10 Shift On |
| N | 4E | | 11 Shift On |
| O | 4F | | 12 Shift On |
| P | 50 | | 13 Shift On |
| Q | 51 | | 14 Shift On |
| R | 52 | | 15 Shift On |
| S | 53 | | 16 Shift On |
| T | 54 | | 17 Shift On |
| U | 55 | | 18 Shift On |
| V | 56 | | 19 Shift On |

| | | | |
|---|---|---|---|
| W | 57 | | 1A Shift On |
| X | 58 | | 1B Shift On |
| Y | 59 | | 1C Shift On |
| Z | 5A | | 1D Shift On |
| [ | 5B | | 2F |
| \ | 5C | | 31 |
| ] | 5D | | 30 |
| ^ | 5E | | 23 Shift On |
| _ | 5F | | 2D Shift On |
| ` | 60 | | 35 |
| a | 61 | | 04 |
| b | 62 | | 05 |
| c | 63 | | 06 |
| d | 64 | | 07 |
| e | 65 | | 08 |
| f | 66 | | 09 |
| g | 67 | | 0A |
| h | 68 | | 0B |
| i | 69 | | 0C |
| j | 6A | | 0D |
| k | 6B | | 0E |
| l | 6C | | 0F |
| m | 6D | | 10 |
| n | 6E | | 11 |
| o | 6F | | 12 |
| p | 70 | | 13 |
| q | 71 | | 14 |
| r | 72 | | 15 |
| s | 73 | | 16 |
| t | 74 | | 17 |
| u | 75 | | 18 |
| v | 76 | | 19 |
| w | 77 | | 1A |
| x | 78 | | 1B |
| y | 79 | | 1C |
| z | 7A | | 1D |
| { | 7B | | 2F Shift On |
| \| | 7C | | 31 Shift On |
| } | 7D | | 30 Shift On |
| ~ | 7E | | 35 Shift On |
| DEL | 7F | | 2A |
| F1 | 81 | \f1 | 3A |
| F2 | 82 | \f2 | 3B |
| F3 | 83 | \f3 | 3C |
| F4 | 84 | \f4 | 3D |
| F5 | 85 | \f5 | 3E |
| F6 | 86 | \f6 | 3F |
| F7 | 87 | \f7 | 40 |
| F8 | 88 | \f8 | 41 |
| F9 | 89 | \f9 | 42 |

| | | | |
|---|---|---|---|
| F10 | 8A | \fa | 43 |
| F11 | 8B | \fb | 44 |
| F12 | 8C | \fc | 45 |
| Home | 8D | \home | 4A |
| End | 8E | \end | 4D |
| → | 8F | \right | 4F |
| ← | 90 | \left | 50 |
| ↑ | 91 | \up | 52 |
| ↓ | 92 | \down | 51 |
| PgUp | 93 | \pgup | 4B |
| PgDn | 94 | \pgdn | 4E |
| Tab | 95 | \tab | 2B |
| bTab | 96 | \btab | 2B Shift On |
| Esc | 97 | \esc | 29 |
| Enter | 98 | \enter | 28 |
| Num_Enter | 99 | \num_enter | 58 |
| *Delete* | 9A | \del | 4C |
| Insert | 9B | \ins | 49 |
| Backspace | 9C | \bs | 2A |
| SPACE | 9D | \sp | 2C |
| *Pause* | 9C | \ps | 48 |
| Ctrl+[ | 9F | \ctr1 | 2F Ctrl On |
| Ctrl+] | A0 | \ctr2 | 30 Ctrl On |
| Ctrl+\ | A1 | \ctr3 | 31 Ctrl On |
| Left_Ctrl_Break | A2 | \l_ctrl_bk | Clear Ctrl Flag |
| Left_Ctrl_Make | A3 | \l_ctrl_mk | Set Ctrl Flag for following char(s) |
| Left_Shift_Break | A4 | \l_shift_bk | Clear Shift Flag |
| Left_Shift_Make | A5 | \l_shift_mk | Set Shift Flag for following char(s) |
| Left_Windows | A6 | \l_windows | E3 (left GUI) |
| Left_Alt_Break | A7 | \l_alt_bk | Clear Alt Flag |
| Left_Alt_Make | A8 | \l_alt_mk | Set Alt Flag for following char(s) |
| Right_Ctrl_Break | A9 | \r_ctrl_bk | Clear Ctrl Flag |
| Right_Ctrl_Make | AA | \r_ctrl_mk | Set Ctrl Flag for following char(s) |
| Right_Shift_Break | AB | \r_shift_bk | Clear Shift Flag |
| Right_Shift_Make | AC | \r_shift_mk | Set Shift Flag for following char(s) |
| Right_Windows | AD | \r_windows | E7 (right GUI) |
| Right_Alt_Break | AE | \r_alt_bk | Clear Alt Flag |
| Right_Alt_Make | AF | \r_alt_mk | Set Alt Flag for following char(s) |
| Num_Lock | B0 | \num_lock | 53 |
| Num_0 | B1 | \num0 | 62 Num Lock On |
| Num_1 | B2 | \num1 | 59 Num Lock On |
| Num_2 | B3 | \num2 | 5A Num Lock On |
| Num_3 | B4 | \num3 | 5B Num Lock On |
| Num_4 | B5 | \num4 | 5C Num Lock On |
| Num_5 | B6 | \num5 | 5D Num Lock On |
| Num_6 | B7 | \num6 | 5E Num Lock On |
| Num_7 | B8 | \num7 | 5F Num Lock On |
| Num_8 | B9 | \num8 | 60 Num Lock On |
| Num_9 | BA | \num9 | 61 Num Lock On |
| Num_Home | BB | \num_home | 5F |

| | | | |
|---|---|---|---|
| Num_PageUp | BC | \num_pgup | 61 |
| Num_PageDown | BD | \num_pgdn | 5B |
| Num_End | BE | \num_end | 59 |
| Num_↑ | BF | \num_up | 60 |
| Num_→ | C0 | \num_right | 5E |
| Num_↓ | C1 | \num_down | 5A |
| Num_← | C2 | \num_left | 5C |
| Print_Scrn | C3 | \prt_sc | 46 |
| System_Request | C4 | \sysrq | 9A |
| Scroll_Lock | C5 | \scroll | 47 |
| Pause | C6 | \menu | 76 |
| Break | C7 | \break | |
| Caps_Lock | C8 | \caps_lock | 39 |
| Num_/ | C9 | \num_/ | 54 |
| Num_* | CA | \num_* | 55 |
| Num_- | CB | \num_- | 56 |
| Num_+ | CC | \num_+ | 57 |
| Num_. | CD | \num_. | 63 Num Lock On |
| Num_DEL | CE | \num_del | 63 |
| Num_INS | CF | \num_ins | 62 |
| Delay_100ms | D0 | \delay | Delay 100 ms |

## 15.1. Appendix C: Ctrl or Alt Output

Table of Ctrl or Alt output for non-printable characters:

| ASCII Code | Control Code | Alt Code |
|---|---|---|
| **SendOptionID** | **Bit 3: 0** | **Bit 3: 1** |
| **00:** | Ctrl-2 | Alt-000 |
| **01:** | Ctrl-A | Alt-001 |
| **02:** | Ctrl-B | Alt-002 |
| **03:** | Ctrl-C | Alt-003 |
| **04:** | Ctrl-D | Alt-004 |
| **05:** | Ctrl-E | Alt-005 |
| **06:** | Ctrl-F | Alt-006 |
| **07:** | Ctrl-G | Alt-007 |
| **08:** | BS | Alt-008 |
| **09:** | Tab | Alt-009 |
| **0A:** | Ctrl-J | Alt-010 |
| **0B:** | Ctrl-K | Alt-011 |
| **0C:** | Ctrl-L | Alt-012 |
| **0D:** | Enter | Alt-013 |
| **0E:** | Ctrl-N | Alt-014 |
| **0F:** | Ctrl-O | Alt-015 |
| **10:** | Ctrl-P | Alt-016 |
| **11:** | : Ctrl-Q | Alt-017 |
| **12:** | Ctrl-R | Alt-018 |
| **13:** | Ctrl-S | Alt-019 |
| **14:** | Ctrl-T | Alt-020 |

| | | |
|---|---|---|
| **15:** | Ctrl-U | Alt-021 |
| **16:** | Ctrl-V | Alt-022 |
| **17:** | Ctrl-W | Alt-023 |
| **18:** | Ctrl-X | Alt-024 |
| **19:** | Ctrl-Y | Alt-025 |
| **1A:** | Ctrl-Z | Alt-026 |
| **1B:** | ESC | Alt-027 |
| **1C:** | Ctrl-\ | Alt-028 |
| **1D:** | Ctrl-] | Alt-029 |
| **1E:** | Ctrl-6 | Alt-030 |
| **1F:** | Ctrl-- | Alt-031 |

## 15.2. Appendix C: Terms and Abbreviations

| | |
|---|---|
| **AAMVA** | American Association of Motor Vehicle Administration |
| **ABA** | American Banking Association |
| **AES** | Advanced Encryption Standard |
| **ASIC** | Application Specific Integrated Circuit |
| **BPI** | Bits per Inch |
| **CADL** | California Drivers License Format (obsolescent) |
| **CE** | European Safety and Emission approval authority |
| **COM** | Serial Communication |
| **CTS** | Clear-To-Send |
| **CDC** | USB to serial driver (Communication Device Class) |
| **DES** | Data Encryption Standard |
| **DUKPT** | Derived Unique Key Per Transaction |
| **DMV** | Department of Motor Vehicle |
| **GND Signal Ground** | Signal Ground |
| **HID** | Human Interface Device |
| **IPS** | Inches per Second |
| **ISO** | International Organization for Standardization |
| **JIS** | Japanese Industrial Standard |
| **JPOS** | Java for Retail Point Of Sale |
| **KB** | Keyboard |
| **KSN** | Key Serial Number |
| **LED** | Light Emitting Diode |
| **LRC** | Longitudinal Redundancy Check Character. |
| **MAC** | Message Authentication Code |
| **MSR** | Magnetic Stripe Reader |
| **OLE** | Object Linking and Embedding |
| **OPOS** | OLE for Retail Point Of Sale |
| **OTP** | One Time Programmable |
| **PAN** | Primary account number |
| **PCI** | Payment Card Industry |
| **PID** | USB Product ID |
| **POS Point of Sale** | Point of Sale |
| **PPMSR** | Serial Port Power Magstripe Reader |
| **P/N** | Part Number |

| | |
|---|---|
| **PS/2 IBM** | Personal System/2 Keyboard Interface |
| **RTS** | Request to Send |
| **SPI** | Serial Peripheral Interface |
| **T1, T2, T3** | Track 1 data, Track 2 data, Track 3 data |
| **TDES** | Triple Data Encryption Standard |
| **VID** | USB Vendor ID |