USER MANUAL

SecureKey™ M100/M130
Encrypted Keypad with Optional
Encrypted MSR

80120502-001-B
June 22, 2011
Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>First draft release for internal review</td>
<td>03/14/11</td>
</tr>
<tr>
<td>A</td>
<td>Initial Release</td>
<td>05/14/11</td>
</tr>
</tbody>
</table>
| B        | -Modified output format and added example data  
          -Added instruction to change the initial key in the demo software  
          -Modified commands to change XML output field settings | 06/22/11   |

Table of Contents

1.0 Introduction .............................................................................................................. 3
2.0 Product Configurations ............................................................................................ 3
3.0 Features .................................................................................................................... 3
4.0 Definition of Terms & Applicable Documents........................................................ 4
5.0 Function & Operation .............................................................................................. 4
6.0 Data Output Format ................................................................................................. 6
   6.1 ID TECH Standard Data Output Format .............................................................. 6
   6.2 XML Data Output Format .................................................................................. 10
7.0 MSR Settings ......................................................................................................... 13
   7.1 Setting Command ............................................................................................... 13
   7.2 Get Setting .......................................................................................................... 13
   7.3 Security Management ......................................................................................... 13
   7.4 Encryption Management .................................................................................... 14
   7.4.1 Key Management ............................................................................................ 15
   7.5 Check Card Format ............................................................................................ 17
   7.6 MSR Data Masking ............................................................................................ 18
8.0 MSR Decryption Demo Software .......................................................................... 19
   The demo software uses the IDTECH demo key ....................................................... 20
   8.1 Card Swipe Data, IDTECH Original Encryption Format .................................. 20
   8.2 Key in data, IDTECH Format ............................................................................ 23
   8.3 Card Swipe Data, XML Format .......................................................................... 24
   8.4 Key in Data, XML Format .................................................................................. 25
9.0 Specifications ......................................................................................................... 28
**1.0 Introduction**

ID TECH SecureKey M series is an encrypted numeric keypad with an optional Magnetic Swipe Reader (MSR). The Secure keypad allows the retailers to not only encrypt credit card data at the magnetic readers but it also encrypts manually entered credit card number. The SecureKey M series has 15 keys (10 Numeric, 5 functional) with a 2x20 backlit LCD.

SecureKey M series keypads encrypt the data using TDES or AES algorithm format with DUKPT key management. For encrypted card reader settings and operations, please refer to 80096504-001 SecureMag User Manual.

SecureKey M series is available in USB-Keyboard and USB-HID interface.

**2.0 Product Configurations**

SecureKey M100/M130 comes in the following configurations:

<table>
<thead>
<tr>
<th>Model Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDKE-504800B</td>
<td>Securekey M100;USB-KB;15Key;No MSR;Blk</td>
</tr>
<tr>
<td>IDKE-534833B</td>
<td>Securekey M130;USB-KB;15Key;3T;Blk</td>
</tr>
<tr>
<td>IDKE-534833BE</td>
<td>Securekey M130;USB-KB;15Key;3T;Blk;Enhanced output</td>
</tr>
<tr>
<td>IDKE-504800BL</td>
<td>Securekey M100;USB-KB;15Key;No MSR;Blk;XML output</td>
</tr>
<tr>
<td>IDKE-534833BL</td>
<td>Securekey M130;USB-KB;15Key;3T;Blk;XML output</td>
</tr>
</tbody>
</table>

**3.0 Features**

- Encrypted numeric keypad with 2x20 LCD and optional encrypted MSR
- 1,000,000 swipe, industry proven Magnetic Stripe Reader
- 20,000,000 key operations for each key
- Meets FCC Class B & CE regulatory requirements
- Plug-n-Play operation for USB-Keyboard and USB-HID interface
- Keypad is encrypted using DUKPT and TDES/AES encryption.
- Optional encrypted MSR with DUKPT and TDES/AES encryption
- Works with Windows 95/98, WINME 2000, XP, & Vista
- Available in standard and XML output format
4.0 Definition of Terms & Applicable Documents

ANSI  American National Standard Institute
ESD    Electrostatic Discharge
HOST   A Personal Computer or Similar Computing Device
ISO    International Standards Organization
MTBF   Mean Time Between Failures
RoHS   Restrictions of Hazardous Substances
USB    Universal Serial Bus

ISO/IEC 7813 – Identification cards, Physical Characteristic
ISO/IEC 7811 – Identification cards, Recording Techniques, Magnetic Stripe
Keyboard Key Code Specification Revision 1.3a, 3/16/2000, Microsoft Corporation

80096504-001 SecureMag User Manual

5.0 Function & Operation

On power_on the device will go into its data capture mode. In data capture mode the device will prompt the user to enter data.

The device would display “Key is not injected!” if the device is not key-injected with encryption enabled and a key is pressed. The evaluation unit is injected with the ID TECH demo key by default and the data can be decrypted using the ID TECH SecureKey demo software.

Function Keys Operation:

Clear:
- Pressing the “Clear” key allows users to remove all entered data at the current level. The current transaction would not be cancelled.

BS:
- Pressing the “BS” (backspace) key allows users to remove the entered data one character at a time.

#Admin:
- Pressing the “#Admin” key when the screen displays “Swipe or Hand-Key Card Number” or “Enter Card Number then press Enter” allows user to enter the Admin Menu. Pressing the “#Admin” key in other screens puts the device in the Help Mode.

Cancel:
- Pressing the “Cancel” key once allows users to remove all the input in the current as well as the previous level. The device then goes back to the previous prompt of the current transaction. If the “Cancel” key is pressed twice, the current transaction would be cancelled and the device goes back to the initial mode.

Admin Menu
When the “Admin” key is pressed, the screen will display "Select manual config 1-5" to prompt the user to select one of five manual entry modes.

Manually-Keyed Configuration Options
Configuration #1: Card Number, Expiration Date
Configuration #2: Card Number, Expiration Date, Zip Code
Configuration #3: Card Number, Expiration Date, Street Number of the Address, Zip Code
Configuration #4: Card Number, Expiration Date, Zip Code, Security Code
Configuration #5: Card Number, Expiration Date, Address, Zip Code, Security Code

When the user selects the key corresponding to a manual mode, and then selects enter, the mode will be configured and the unit will return to the data capture mode. If the user selects more than one key, then the last key selected will be used to select the mode.
If a invalid key is selected the unit will display "error" then "Select manual config 1-5"

Help Mode
If the user selects the Admin key while in Admin mode, the unit enters the Help Mode. In the Help Mode, the unit displays short text messages of the various manual entry configurations with a 3 seconds pause between each message. Hitting any key in the Help Mode makes the unit return to the Admin Menu.
6.0 Data Output Format

There are two data output formats for SecureKey M100/ M130, one is the ID TECH standard data output format and the other is XML output format.

6.1 ID TECH Standard Data Output Format

STX
1. Data Length low byte
2. Data Length high byte
3. Card Encode Type
4. Track 1-3 Status
5. T1 data length
6. T2 data length
7. T3 data length
8. Clear/mask data sent status (not applicable if key-in)
9. Encrypted/Hash data sent status (not applicable if key-in)
10. T1 clear/mask data - (Track 1 data) (not applicable if key-in)
    T2 clear/mask data - (Track 2 data) (not applicable if key-in)
    T3 clear/mask data - (Track 3 data) (not applicable if key-in)
    T1 encrypted data - (Track 1 encrypted data) (not applicable if key-in)
    T2 encrypted data - (Track 2 encrypted data) (not applicable if key-in)
    T3 encrypted data - (Track 3 encrypted data) (not applicable if key-in)
AddrStatus (1 byte)
ECData - Encrypted card data (max: 180 bytes) (for SecureKey key-in only)
    ECDData - hashed (20 bytes each) (for SecureKey key-in only)
Exp - Expiration date (len: 4+1 bytes) (for SecureKey key-in only)
AVSAAddr - Street number (max: 20+1 bytes) (for SecureKey key-in only)
AVSZip - Zip code (max: 10+1 bytes) (for SecureKey key-in only)
T1 hashed (20 bytes each) (if encrypted and hash tk1 allowed) (not applicable if key-in)
T2 hashed (20 bytes each) (if encrypted and hash tk2 allowed) (not applicable if key-in)
T3 hashed (20 bytes each) (if encrypted and hash tk3 allowed) (not applicable if key-in)
KSN (10 bytes)
CheckLRC
CheckSum
ETX

Note 1 : Card Encode Type

<table>
<thead>
<tr>
<th>Value</th>
<th>Encode Type Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 / 80</td>
<td>ISO/ABA format</td>
</tr>
</tbody>
</table>
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/mask 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

Other note:
- Data Length low byte/high byte should be in length of characters (USBKB) and in length of bytes (USBHID).
- ECData include encrypted card key-in data, expiration date (YYMM) and 3/4 digit security code (cc).
  The format should be:
1) (Security level 3) Card Data=YYMM=cc
2) (Security level 3 without cc) Card Data=YYMM=
3) (Security level 4) Card Data=YYMM=cc= Session ID (8 bytes)

Each field is separated by delimiter ‘=’, this should always present even cc is not keyed-in.

- The format of the field ECData, Exp, AVSAddr and AVSZip should be:

<table>
<thead>
<tr>
<th>1 byte length of decrypted data in Hex (in byte)</th>
<th>Data</th>
</tr>
</thead>
</table>

The length byte describe the length of decrypted data not include sessionID if it is level 4.

- If the field is not applicable then the field will not be sent out.
- AddrStatus: eg. If bit 2 is 1, expDate exist.

<table>
<thead>
<tr>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>nil</td>
<td>expDate</td>
<td>AVSAddr</td>
<td>AVSZip</td>
</tr>
</tbody>
</table>

**Original Encryption Format Swipe Output**

028801001F372300%*5150********7903^PAYPASS/MASTERCARD^**************
****?*;5150********7903=***************?*8871B640F379F3BD8D057A13F81454
39B28D80BE8A43F3440D85928F576065EEE1BA54CAADFF67D552C20CBF1A9F
34B63402B967998FC7C80487C8A6DBFD46975985D3D7E865FEEF6A48930751DC9
71FDFCBC1989294B7EF6F0D0007AA731C31F574608EB85E57751ADA48970F96B0E
8BECDB94D672D746C2CC75176FA6E0C9E6FEFE0B154A0959B6299490125000000
00197F6903

Key Value: F5 BF 6B E8 55 AB 92 3A DE 7E 77 40 D8 46 F9 DE
KSN: 62 99 49 01 25 00 00 00 00 1A

Decrypted Data:

Data in ASCII Format
%5B150710200107903^PAYPASS/MASTERCARD^090910140000631?;5150710200
107903=090910140000631?0

Data in HEX Format
25423531353037313032303031303739303335E50415950415332F4D415354455243415
2445E30393039313031343030303633313F3F3B3533135307331303230303130373930
333D30393039313031343030303633313F30000000000000

**Enhanced Encryption Format Swipe Output**

028C01801F372300039B%*5150********7903^PAYPASS/MASTERCARD^**************
****?*;5150********7903=***************?*C5E7500896207CBFC9B1DA1
9F6EFFB392E26C04C3BC76121C480A3B6FC122EDCE85B813682DAC3628002507
B424B31A0D6196BDF563F1821470553DF7F5CB7EA2226764915B3A1B4119010513
Key Value: 32 68 28 A3 E4 F5 84 48 09 D2 8A B5 EB B8 AA 74
KSN: 62 99 49 01 25 00 00 00 00 1C

Decrypted Data:

Data in ASCII Format
%B5150710200107903^KSN: 62 99 49 01 25 00 00 00 00 1C
Decrypted Data:

Data in HEX Format
25423531353037313032303033303739303335E504159504153532F4D415354455243415
2445E303930393130334030303633313F3F00
3B353135303731303230303330373930333D303930393130334030303633313F300
000000000

Manual key in
029C00850000000718A1F6300C7241C9933DE31A01AB0C6021563FFC7B4810D9
4DA8863CE5EC84B37EA79A87D96572047CFCF1068F0430393039053130373231053
93036333062994901250000000001D095B03

Key Value: B8 C7 3E 0A 17 58 09 5 A 7 A 86 44 6F 9B B5 76 FF
KSN: 62 99 49 01 25 00 00 00 00 1D

Decrypted Data:

Data in ASCII Format
515710200107903=90910140000631??
;5150710200107903=90910140000631?0

Data in HEX Format
25423531353037313032303033303739303335E504159504153532F4D415354455243415
2445E303930393130334030303633313F3F00
3B353135303731303230303330373930333D303930393130334030303633313F300
000000000

Manual key in
029C00850000000718A1F6300C7241C9933DE31A01AB0C6021563FFC7B4810D9
4DA8863CE5EC84B37EA79A87D96572047CFCF1068F0430393039053130373231053
93036333062994901250000000001D095B03

Key Value: B8 C7 3E 0A 17 58 09 5 A 7 A 86 44 6F 9B B5 76 FF
KSN: 62 99 49 01 25 00 00 00 00 1D

Decrypted Data:

Data in ASCII Format
515710200107903=90910140000631??
;5150710200107903=90910140000631?0

Data in HEX Format
353135303731303230303330373930333D303930393130334030303633313F300
000000000
6.2 XML Data Output Format

The XML data output format is as below. Messages (swiped and keyed credit, debit, other, gift, drivers licenses, etc.) need to include at least the <Addr></Addr> tag. The XML tags needs to be in the following order:

```xml
<DvcMsg Ver="1.1">
    <Dvc attribute list .../></Dvc>
    <Card attribute list .../></Dvc>
    <Addr attribute list .../></Dvc>
    <Tran attribute list .../></Dvc>
</DvcMsg>
```

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Attribute</th>
<th>Required</th>
<th>Max Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ver</td>
<td>DvcMsg</td>
<td>Required</td>
<td>10</td>
<td>String</td>
<td>Device Message Version (use 1.1)</td>
</tr>
<tr>
<td>App</td>
<td>Dvc</td>
<td>Required</td>
<td>50</td>
<td>String</td>
<td>Application Name</td>
</tr>
<tr>
<td>AppVer</td>
<td>Dvc</td>
<td>Required</td>
<td>10</td>
<td>String</td>
<td>Application Version</td>
</tr>
<tr>
<td>DvcType</td>
<td>Dvc</td>
<td>Required</td>
<td>40</td>
<td>String</td>
<td>Device Type (MODEL- MANUFACTURER)</td>
</tr>
<tr>
<td>DvcSN</td>
<td>Dvc</td>
<td>Required</td>
<td>40</td>
<td>String</td>
<td>Device Serial Number</td>
</tr>
<tr>
<td>Entry</td>
<td>Dvc</td>
<td>Required</td>
<td>20</td>
<td>String</td>
<td>Card Entry Method (SWIPE, MANUAL, CONTACTLESS)</td>
</tr>
<tr>
<td>CEncode</td>
<td>Card</td>
<td>Optional</td>
<td>2</td>
<td>Integer</td>
<td>Card Encoding Type: 0 = ISO/ABA, 1 = AA/MVA, 2 = Keyed (Manual Keyed), 3 = Other</td>
</tr>
<tr>
<td>Trk1</td>
<td>Card</td>
<td>Optional</td>
<td>240</td>
<td>String</td>
<td>Track 1 (currently only used for non-financial cards)</td>
</tr>
<tr>
<td>Trk2</td>
<td>Card</td>
<td>Optional</td>
<td>180</td>
<td>String</td>
<td>Track 2 (currently only used for non-financial cards)</td>
</tr>
<tr>
<td>Trk3</td>
<td>Card</td>
<td>Optional</td>
<td>180</td>
<td>String</td>
<td>Track 3 (currently only used for non-financial cards)</td>
</tr>
<tr>
<td>ETrk1</td>
<td>Card</td>
<td>Optional</td>
<td>240</td>
<td>String</td>
<td>Encrypted Track 1</td>
</tr>
<tr>
<td>ETrk2</td>
<td>Card</td>
<td>Optional</td>
<td>180</td>
<td>String</td>
<td>Encrypted Track 2</td>
</tr>
<tr>
<td>EData</td>
<td>Card</td>
<td>Optional</td>
<td>180</td>
<td>String</td>
<td>Encrypted Card Data (Card Number−ExpDate(YYMM)−Security Code)</td>
</tr>
<tr>
<td>CDataKSN</td>
<td>Card</td>
<td>Optional</td>
<td>40</td>
<td>String</td>
<td>Card Data Key Serial Number</td>
</tr>
<tr>
<td>MskPAN</td>
<td>Card</td>
<td>Optional</td>
<td>30</td>
<td>String</td>
<td>Masked PAN. Format: 4003*******6781</td>
</tr>
<tr>
<td>Exp</td>
<td>Card</td>
<td>Optional</td>
<td>8</td>
<td>String</td>
<td>Expiration Date. Format: YYMM</td>
</tr>
<tr>
<td>CHolder</td>
<td>Card</td>
<td>Optional</td>
<td>80</td>
<td>String</td>
<td>Cardholder Name</td>
</tr>
<tr>
<td>AVSAddr</td>
<td>Addr</td>
<td>Optional</td>
<td>50</td>
<td>String</td>
<td>AVS Address</td>
</tr>
<tr>
<td>AVSZip</td>
<td>Addr</td>
<td>Optional</td>
<td>20</td>
<td>String</td>
<td>AVS Zip Code</td>
</tr>
<tr>
<td>TranType</td>
<td>Tran</td>
<td>Required</td>
<td>40</td>
<td>String</td>
<td>Transaction Type (CREDIT, DEBIT)</td>
</tr>
</tbody>
</table>
The data output format is XML output message protocol.

The DvcType, DvcApp, DvcMsgVer, and AppVer field can be configured by the following commands:

53 77 53 4B <function ID><data length> <data>

Set DvcType example: 53 77 53 4B 5C 0B 4D 31 33 30 2D 49 44 54 45 43 48

Set DvcApp example: 53 77 53 4B 5D 12 53 65 63 75 72 65 4B 65 67 79 20 53 6F 66 74 77 61 72 65

Set DvcMsgVer example: 53 77 53 4B 5E 03 31 2E 30

Set AppVer example: 53 77 53 4B 5F 03 31 2E 30

Credit Card Swipe Sample XML:

```xml
<DvcMsg Ver="1.1"><Dvc App="SecureKey Software" AppVer="1.0" DvcType="M130-IDTECH" DvcSN="FFFFFFFFFFFFFFFFF"
Entry="SWIPE"></Dvc><Card CEncode="0" ETrk1="9719BCB11786D9F5D26CD2350C6307D82FA980E673A02760F2383C2AF9BB8A6A875083B049582C91FCB542A06591DF223034C1A9EAC64A3166406B8516123F5200AC773BAF8ECDD"
ETrk2="4623A11A24D344A71137EB2EE5A2E5F4A013E7D286FB9A8A5523316720DF6B47473166171154A07F" CDatasKSN="62994901230000000002F" Exp="0809"
MskPAN="4266*******9999" CHolder="BUSH JR/GEORGE W.MR" EFormat="4"></Card><Tran TranType="CREDIT"></Tran></DvcMsg>
```

CarriageReturn

Key Value: ED 07 9C 5F 5E 5D F7 E2 03 7B 7F 36 F3 10 54
KSN: 62 99 49 01 23 00 00 00 00 02F

Decrypted Data in ASCII:
Decrypted Data in Hex:
254234323638343130383839393939393F42555348204A522F47454F5247452057E3038303931303030303030303030303630303F
3B343236383431303838393939393D30383039313030303030303034363F3000000000

Credit Card Manually Keyed Sample XML:

```xml
<DvcMsg Ver="1.1"><Dvc App="SecureKey Software" AppVer="1.0"
DvcType="M130-IDTECH" DvcSN="FFFFFFFFFFFFFF"
Entry="MANUAL"></Dvc><Card CEncode="2"
ECDATA="F4EA319F165989392A51A747EF82FF2461DC3CB8B68995F315FCFE5
4A81CF6" CDataKSN="62994901230000000030" Exp="1206"
MskPAN="1234********6789" EFormat="4"></Card><Addr AVSAddr="10721"
AVSZip="91741"></Addr><Tran TranType="CREDIT"></Tran></DvcMsg>
```

CarriageReturn

Key Value: CA DC 1C 5A D6 5A FF 5D 06 81 A1 E3 37 51 A4 5A
KSN: 62 99 49 01 23 00 00 00 00 30

Decrypted Data in ASCII:
1234567890123456789=1206=123

Decrypted Data in Hex:
313233343536373839303132333435363738393D313230363D313233000000

Non-Financial Card Swipe Sample XML:

```xml
<DvcMsg Ver="1.1"><Dvc App="SecureKey Software" AppVer="1.0"
DvcType="M130-IDTECH" DvcSN="FFFFFFFFFFFFFF"
Entry="MANUAL"></Dvc><Card CEncode="3"
Trk1=""Trk2="Track1ofGiftCardData"
CDataKSN="A08B000C000002000E6" MskPAN="1212********5588" Exp="1512"
CHolder=" BUSH JR/GEORGE W.MR " EFormat="4"></Card><Addr></Addr><Tran
TranType="OTHER"></Tran></DvcMsg>
```

CarriageReturn

Note:
- Manually entered data should always be financial card data.
- The non-encrypted track fields (Trk1, Trk2, Trk3) are only used when the format is a non-financial card.
- "EFormat" is defined by the application

7.0 MSR Settings

7.1 Setting Command
The setting data command is a collection of one or more function setting blocks and its format is as the following:

Command: <STX><S><FuncSETBLOCK1>…<FuncBLOCKn><ETX><LRC>
Response: <ACK> or <NAK> for wrong command (invalid funcID, length or value)

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

<FuncID><Len><FuncData>
The setting command will function with any one, any group or all the setting in one command.

Where:
<FuncID> is one byte identifying the setting(s) for the function.
.Len> is a one byte length count for the following function-setting block <FuncData>.
<FuncData> is the current setting for this function. It has the same format as in the sending command for this function.

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

<FuncID>, <Len> and <FuncData> definition are same as described above.
Note: ReviewID (value 0x1F) will return all funcID-s.

7.3 Security Management
The MSR reader is intended to be a secure reader. Security features include:
- Can include Device Serial Number
- Can encrypt track 1 and track 2 data for all bank cards (ETrk1 and ETrk2 will be empty if non bank card is swiped).
- Provides clear text confirmation data including card holder’s name and a portion of the PAN as part of the Masked Track Data (for bank cards)
- Optional display expiration data (for bank cards)
- Configurable Security Level
The reader supports five Security Levels. This allows customer to select the security profile needed for the application. The Security Level can be raised by command, but can never be lowered:

- **Level 0**
  Security Level 0 is a special case. It signifies that all DUKPT keys have been used. In this case the unit is at the end of its useful life. This level is set automatically by the reader when it runs out of DUKPT keys. The life time of DUKPT keys is one millions. Once reach the end of keys’ life time, user should inject DUKPT keys again.

- **Level 1**
  Reader properties are as configured from factory having the lowest level of default settings. There is no encryption process, no key serial number transmitted with decoded data. The reader has read operation and decoded track data is sent in default format. Encrypt type TDES and AES cannot be selected under Level 1.

- **Level 2**
  Key Serial Number and/or Initially Loaded Device Key have been injected. The encryption process is not activated and decoded track data is sent in default format. Key Serial Number and Initially Loaded Device Key can be set only once after manufacture.

- **Level 3**
  Both Key Serial Number and Initially Loaded Device Keys are injected and encryption is on. The encryption process is activated. The output of level 3 will be different from level 1 and level 2. Clear data output cannot be selected under Level 3. The output format in this level is more rigidly fixed so many track formatting output options are not supported, see function ID table for limitations.

- **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 four byte Message Authentication Code (MAC) at the end. Note that 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, this could be retrieved using Get DUKPT KSN and Counter command. The output format in this level is more rigidly fixed so many track formatting output options are not supported, see function ID table for limitations.

### 7.4 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.

### 7.4.1 Key Management

The encryption key is TDES with 128 bit keys (128 bit keys including parity) or AES encryption with double length keys.

**Key Injection**

As this device is using DUKPT as key management, it is necessary to load initially the Key Serial Number (KSN) and the Loaded Device Key before transaction.

SecureMag is designed to support multiple key injection events, which means KSN and Initially Loaded Device Key can be loaded more than one time. Key injection commands are “Get Key status”, “Load KSN” and “Load Device Key”. In order to keep the security of key transportation, “Load Device Key” command is encrypted using RSA public key. Command protocol is the same, the only difference is <Command Data> and <Respond Data> are encrypted with AES256 and then use BASE64 convert to output data. Customers who want to do their own key injection share this public key.

“Load KSN” and “Load Device Key” are initially injected by secure facility.

KSN and Initially Loaded Device Key loading commands and responses protocol:

**Command:**

<STX><’F’><’F’><Command Data (BASE64)><0x0D><0x0A><ETX><LRC>

**Response:**

<ACK/NAK><STX><’F’><’F’>< Respond Data (BASE64)><0x0D><0x0A><ETX><LRC>

STX: 0x02
ETX: 0x03
ACK: 0x06
NAK: 0x15
BASE64: Data encoded with base64 algorithm
LRC: Xor’d all the data before LRC except STX.

A successful key loading process includes the following steps:
• Enter Key loading mode (optional)
  Command: \55\01\06\08\09\01\5A
  Response: \06

• Get Key status
  Command Data: <FF><13><01><02><LRC>
  Response Data: <FF><00><01><04><LRC>

• Load KSN
  Command Data: <FF><0A><11><KSN#><KSN bytes><LRC>
  Response Data: <FF><00><06><RESPONSE CODE><LRC>
  <KSN#>: TDES: 0x32   DES: 0x0A
  <KSN bytes>: 10 bytes ASCII for KSN
  <RESPONSE CODE>: 6 bytes data in ASCII format which is converted from the first 3 cipher hex data. These cipher data are generated by encrypting KSN bytes and "00 00 00 00 00 00 00 00 00".

• Load Device Key
  Command Data: <FF><0A><LENGTH><KEY#><KEY bytes><LRC>
  Response Data: <FF><00><06><RESPONSE CODE><LRC>
  <LENGTH>: TDES: 0x21   DES: 0x11
  <KEY#>: TDES: 0x33   DES: 0x0B   PUBLIC_N: 0x37
          PUBLIC_E: 0x38
  <KEY bytes>: TDES: 0x20   DES: 0x10
  <RESPONSE CODE>: 6 bytes data in ASCII format which is converted from the first 3 cipher hex data. These cipher data are generated by encrypting KEY bytes and "00 00 00 00 00 00 00 00 00".

  <0x02><0x46><0x46><0x04><0x52><LENGTH_L><LENGTH_H><encrypted base64 block><0x03><lrc>

  <lrc>: Xored from the 1st 0x46 to 0x03
  <Length_L><Length_H> is the length of <encrypted base64 block>, <length_L> is 0x00, <length_H> is 0x01 here.

  <encrypted base64 block>: encrypted key block.
The 1st step is using base64 to wrap DUKPT base key, generate a base64 block:
<0xff><0x0a><0x21><0x33><32 bytes Key ascii code>

The 2nd step: adjust/pad base64 block to generate following block:
<0x00><0x00><0x00><base64block length><0xff>…<0xff><0x00><base64block>
Here is the way to generate this block: set Byte1, byte2, byte3 to 0, byte4 is the length of base64block. Other bytes are padded with 0xff. The byte before base64block is set to 0. Totally there are 256 bytes in this block.

The last step: encrypt the whole 256 bytes block with RSA public key.

- Quit Key loading mode (optional)
  
  Command: `\55\01\06\08\01\01\5A`
  
  Response: `\06`

Example commands:

1. Get Key status
   
   Command: `\02\46\46\2F\78\4D\42\41\75\38\3D\0D\0A\03\LRC`
   
   Response: `\06\02\46\46\..............\0D\0A\03\LRC`

2. Load KSN
   
   Command:
   `\02\46\46\2F\77\6F\52\4D\7A\5A\47\52\6B\59\35\4F\44\63\32\4E\54\51\7A\4D\6A\45\77\52\54\43\69\0D\0A\03\5D`
   
   Response: `\06\02\46\46\..............\0D\0A\03\LRC`

3. Load Encryption Key
   
   Command:
   `\02\46\46\2F\77\6F\68\4D\7A\5A\42\51\7A\49\35\4D\6B\5A\42\51\54\45\7A\4D\54\43\4E\45\51\34\4E\54\68\42\51\6A\4E\42\4D\30\51\33\52\44\55\35\4D\7A\4E\42\6C\51\3D\0D\0A\03\2D`
   
   Response: `\06\02\46\46\..............\0D\0A\03\LRC`

### 7.5 Check Card Format

- ISO/ABA (American Banking Association) Card

  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 checks
  
  Track1 2\textsuperscript{nd} byte is ‘B’.
  There is only one ‘=’ in track 2 and the position of ‘=’ is between 12\textsuperscript{th} ~ 20\textsuperscript{th} character.
  Total length of track 2 is above 19 characters.
  Card number range in PAN will be used to identify bank card.
- **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)**

### 7.6 **MSR Data Masking**

**For ABA Card Data (Card type 0)**
For cards that 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). Optional expiration date may be revealed.
- The rest of the characters should be masked using mask character.

Mask character default value is ‘*’. 
8.0 MSR Decryption Demo Software

A SecureKey demo software is available to demonstrate the MSR data decryption. Please see the below screenshots:

This demo software can be used for USB-HID or USB KB interface. For USB KB interface, please make sure the cursor is placed in the “manual command” window before swiping a card.

The following demo software screenshots are shown for reference and might not reflect the latest demo software version.
The demo software uses the IDTECH demo key
0123456789ABCDEFFEDCBA9876543210
to decrypt the swiped or entered data by default. To change the decryption key, click on “input initial key”.

8.1 Card Swipe Data, IDTECH Original Encryption Format

Type 52 85 on the manual command screen to see the current SecureKey setting and press “Send Command”
Check the 5th byte of the response, if it’s “30”, the SecureKey is in IDTECH original encryption format, for example 06 02 85 01 30 03 85
If the 5th byte is “31”, the SecureKey is in IDTECH enhanced encryption format.

To change the encryption format, go to “MSR Security” and select the original or enhanced encryption format.

Swipe a card, the output and decrypted data will be shown on screen.
8.2 Key in data, IDTECH Format

Manually key in the card data on the device, the data will show on the demo as the following:

![SecureKey USB Demo ver 5.0](image)

**Manual Command / Reader Output**

- **Reader Output (SecureKey Key In Format):**
  - CG56C065C0C0666C0793C0D363E65E337C2B8687F5328C0CC0228D0DDB05E39E216B046637BB0C273A5E53E3E58449
  - EBFDBACE85324B70A6EA41218285B8536DA9409

- **ECD data:**
  - ADF79E83FC07F2E66C03223C0D666E39E2136493F379E0CCAB73A5E800B3544

- **ECD data Hash:**
  - 5BFD6E41E836AE70A6EA41218285B8536DA9409

- **Expiration:**

**Command Output / Decrypted Data**

- **Key Value:** 54 86 38 31 87 7C 5F 72 E3 DD FB 48 57 42 EC 7D
- **ICSN:** 62 99 49 C1 25 00 00 00 00 19

- **Decrypted Data:**
  - Data in ASCII Format: 51071020102303-009=356
  - Data in HEX Format: 353135393731303230303130373330333D3033303330333F3600000000000000

![SecureKey USB HID Reader Connected (IDT Format, TDES Encryption)](image)
8.3 Card Swipe Data, XML Format

Make sure the “Card Swipe” button is checked and then swipe a card.

Press the “decrypt” button, the following data will show

The decrypted data is as shown below:
8.4 **Key in Data, XML Format**

Make sure the “Key In” button is checked and then enter the card data.
Click on the “decrypt” button

Card Data

2EE8A517AFE4096EAAFB62F106FE84008C0E1860A66ED8

OK

KSN

6299901250000000007

OK
# 9.0 Specifications

## Mechanical

<table>
<thead>
<tr>
<th>ITEM</th>
<th>SPECIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyswitch Information</td>
<td></td>
</tr>
<tr>
<td>Total/Pre-Travel</td>
<td>2.5 + 0.5 mm/ 1.5 + 0.4 mm</td>
</tr>
<tr>
<td>Operating Type</td>
<td>Tactile Type</td>
</tr>
<tr>
<td>Operating Force</td>
<td>55 + 7g</td>
</tr>
<tr>
<td>Tactile Feel Force</td>
<td>30 + 14g</td>
</tr>
<tr>
<td>Letter of Keycap</td>
<td>Traditional North American</td>
</tr>
<tr>
<td>Material of Key switch</td>
<td>Silicone Rubber (Rubber Key Pad)</td>
</tr>
<tr>
<td>Keyboard Information</td>
<td></td>
</tr>
<tr>
<td>Enclosure</td>
<td>Top &amp; Bottom Case</td>
</tr>
<tr>
<td>Material</td>
<td>High Impact ABS</td>
</tr>
<tr>
<td>Color</td>
<td>Black</td>
</tr>
<tr>
<td>Cable Information</td>
<td></td>
</tr>
<tr>
<td>Jacket Material</td>
<td>Polyester 0.075 mm</td>
</tr>
<tr>
<td>Conductors</td>
<td>Polyester 0.10 mm</td>
</tr>
<tr>
<td>Color</td>
<td>Upper circuit: 3M467+PET125S</td>
</tr>
<tr>
<td>Length</td>
<td>Lower circuit: 3M467+PET 100S</td>
</tr>
<tr>
<td>PC Connector</td>
<td>Acheson ED-725A 5~10 um</td>
</tr>
<tr>
<td>Keyboard Membrane Material</td>
<td></td>
</tr>
<tr>
<td>Spacer</td>
<td>The auxiliary ports are only on the USB keyboard &amp;</td>
</tr>
<tr>
<td>Back-up Plate</td>
<td>located horizontal to each other on the rear. USB</td>
</tr>
<tr>
<td>Upper Circuit</td>
<td>plastic color is white.</td>
</tr>
<tr>
<td>Lower Circuit</td>
<td></td>
</tr>
<tr>
<td>Silver</td>
<td></td>
</tr>
<tr>
<td>Interface</td>
<td>USB-KB and USB-HID</td>
</tr>
</tbody>
</table>
### Electrical

<table>
<thead>
<tr>
<th>ITEM</th>
<th>SPECIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Rating</td>
<td>+5.0 VDC ±10%, 60ma Max (excludes ICC)</td>
</tr>
<tr>
<td>Type of Circuit</td>
<td>1 Circuit 1 Contact</td>
</tr>
<tr>
<td>Insulation Resistance</td>
<td>DC 100V 50 M Ω Min</td>
</tr>
<tr>
<td>Bounce</td>
<td>10 ms Max</td>
</tr>
<tr>
<td>Operating Life</td>
<td>20,000,000 keystrokes</td>
</tr>
<tr>
<td>Industry Requirements</td>
<td>FCC class B and CE</td>
</tr>
</tbody>
</table>

### Quality & Reliability

<table>
<thead>
<tr>
<th>ITEM</th>
<th>SPECIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI Requirement</td>
<td>The keyboard meets the FCC class B limits</td>
</tr>
<tr>
<td>ESD Immunity</td>
<td>The keyboard passes 0KV to 8 kV minimum without any data loss; passes 8KV to 15 kV minimum that may cause malfunctions. No internal components are destroyed and after reset, the keyboard functions normally.</td>
</tr>
<tr>
<td>MTBF</td>
<td>The main operating time between failures will be more than 60,000 hours</td>
</tr>
<tr>
<td>Drop</td>
<td>610 mm (24&quot;) height</td>
</tr>
<tr>
<td>Vibration</td>
<td>Drop: 4 corner, 4-sidelines, 2-sides front/back</td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>0°C ~ 40°C</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>-20°C ~ + 40°C</td>
</tr>
</tbody>
</table>

### MagStripe Reader

<table>
<thead>
<tr>
<th>Number of tracks</th>
<th>Tracks 1 &amp; 2 or Tracks 2 &amp; 3 or Tracks 1, 2 &amp; 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>TDES or AES with DUKPT key management</td>
</tr>
<tr>
<td>Compatibility</td>
<td>ISO 7810 and 7811-1 through -6</td>
</tr>
<tr>
<td>Output data formatting</td>
<td>Standard or XML output format</td>
</tr>
<tr>
<td>Operating Life</td>
<td>1,000,000 card swipes</td>
</tr>
<tr>
<td>Card speed range</td>
<td>3 to 60 IPS (Inches Per Second)</td>
</tr>
</tbody>
</table>