

**719 and 727  
Mifare Card  
Programmer  
Software User Manual**

**Revision 1.10**



# Table Of Contents

<b>Installation</b>	1
<b>Initial setup</b>	2
<b>Read and writing to Mifare Ultralight cards and NFC tags</b>	5
<b>Reading and writing to Mifare card sectors</b>	7
<b>Keys</b>	10
<b>Sector Trailer</b>	11
<b>Configuration Card for 718-xx readers and 724-xx modules</b>	12
<b>Secure ID</b>	14
<b>Mifare Application Directory</b>	15
<b>Configuration Card for 735-xx/737-xx keyboard wedge readers</b>	16
<b>Mifare Plus Card Personalisation</b>	17
<b>Understanding Mifare Plus</b>	18

# Installation

To install the software insert the CD marked '719-xx Mifare Card Programmer' and run setup.exe from this disk. Follow the instructions on the screen.

Once the installation is complete you need to run 'Mifare Card Programmer' program from the Program Menu.

The first time you go to the 'Keys' screen you will be prompted to enter a pass phrase of not less than 12 characters. The program will use this phrase to encrypt the Mifare keys that you enter when they are stored on the PC. **DO NOT** forget this phrase as you will need it to be able to view and edit your Mifare keys. **There is no way of recovering this phrase if you lose it.**

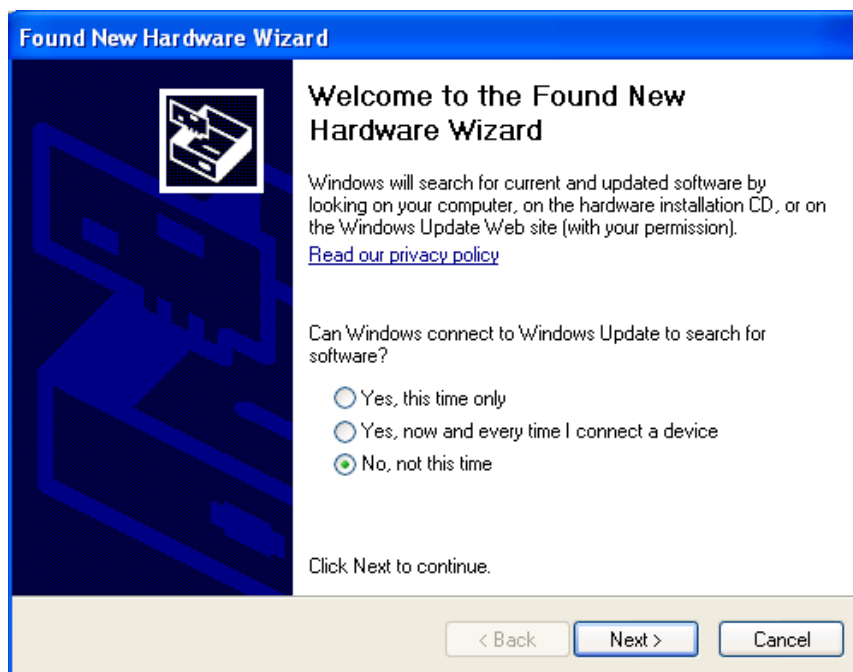
# Initial setup

## Connecting the 719-xx programmer unit

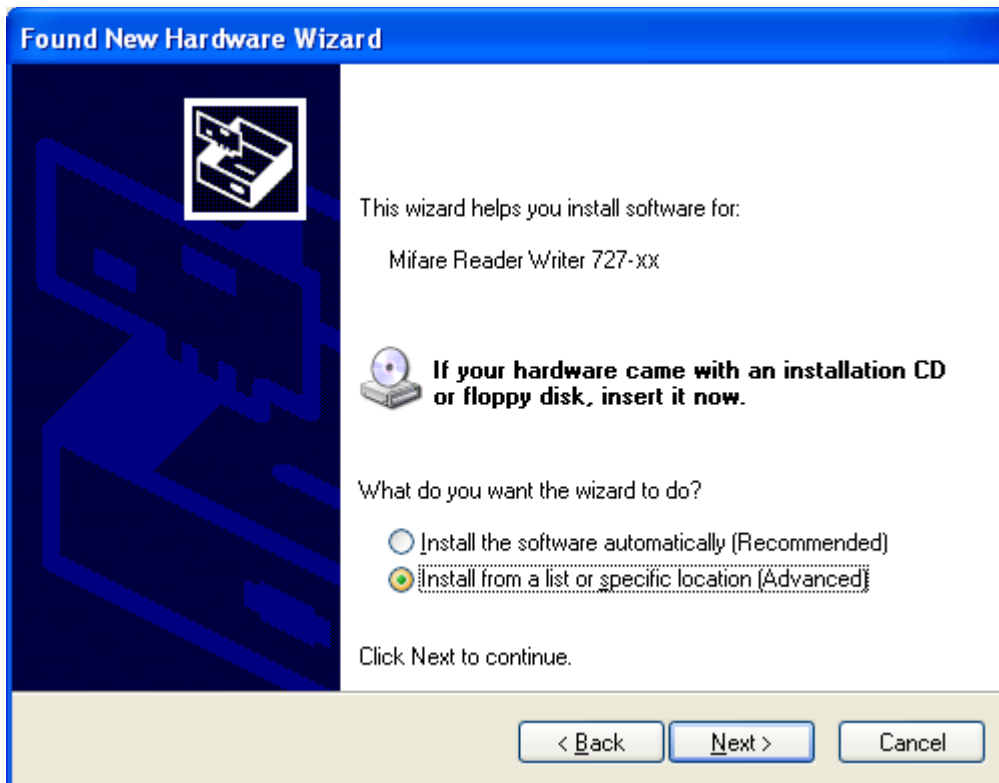
1. Connect the 719-xx programmer to a serial port on your PC using the serial cable provided.
2. Supply 9-13Vdc to the programmer through the DC power jack.
3. In the software go to the 'General' tab and select the serial port the programmer is connected to. On successful connection the current firmware version of the programmer will be displayed.

## Connecting the 727-xx programmer unit

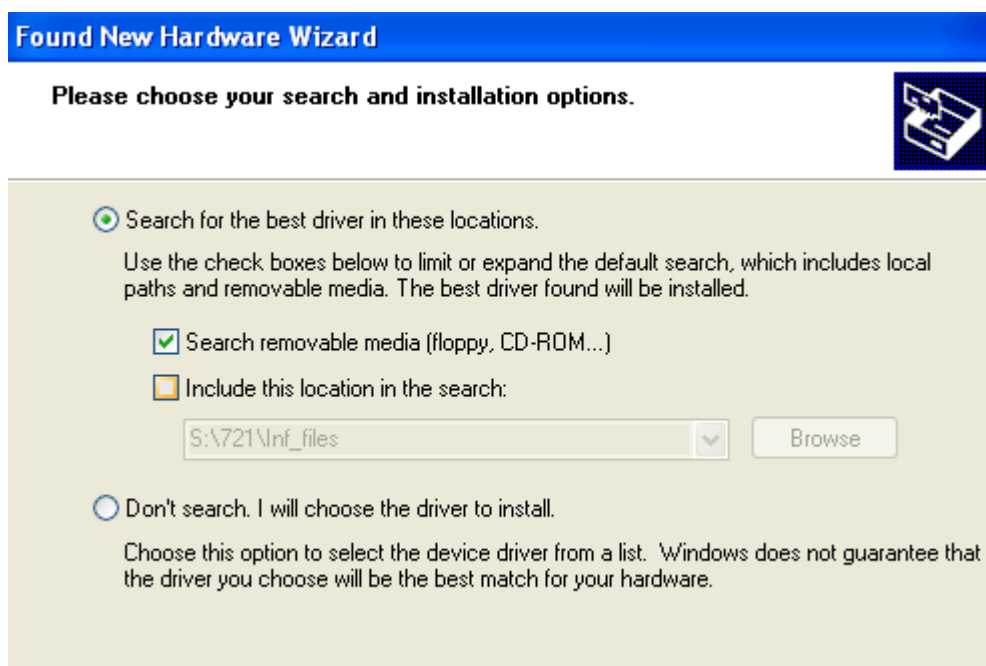
1. Connect the 727-xx programmer unit to a USB port on your PC using the USB cable provided. **It is advisable to connect the programmer to the same USB port every time you use it.** The first time you plug in, the device driver from the CD will need to be installed:
2. After the USB cable is connected, the initial Hardware Wizard dialog appears. Select "No not this time" and click Next.



3. On the next screen select "Install from a list of specific locations" option. Click Next.



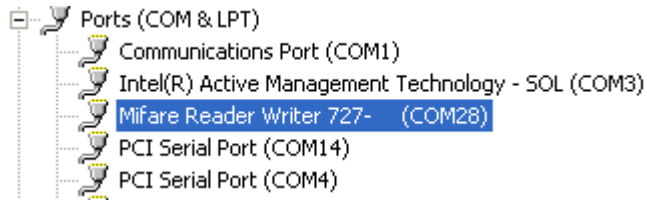
4. At the next screen select “Search for the best driver in these locations and select “Search removable media”. Now click Next.



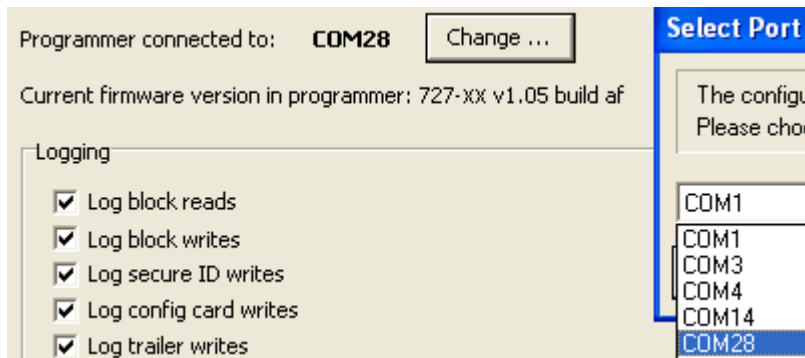
5. If you get a warning that the driver has not passed Windows Logo testing, just select “Continue Anyway” .

6. The final dialog will indicate that the driver has been successfully installed. Click Finish. It will not be necessary to restart the system.

The programmer will appear in the Device Manager view (available from the System applet in the Control Panel) when the installation is complete. The Mifare Reader Writer will appear as shown below. Take note of the COM port number as it will be required when using the 719-xx software.



In the software go to the 'General' tab and select the serial port that was allocated to the programmer. In the above example it is COM28. On successful connection the current firmware version of the programmer will be displayed.

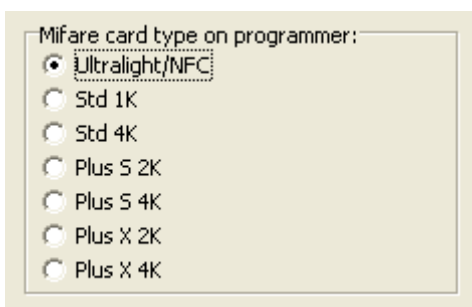


**Note: The programmer must be plugged in before starting the program.**

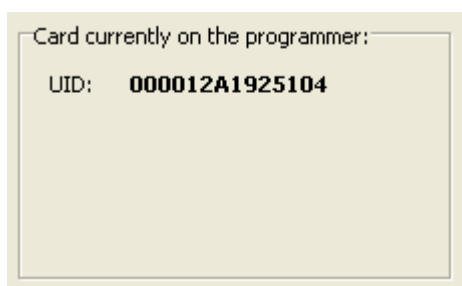
**Note: The program must be closed before unplugging the USB connector.**

# Read and writing to Mifare Ultralight cards and NFC tags

Select the 'User data blocks' tab and select 'Ultralight/NFC' card type.



Place the card on the programmer. The UID will automatically be read and displayed:





The card type is selected from the drop down box, which has options for Ultralight cards and NFC tags, in addition a custom size can be specified by directly entering the page count. The memory is displayed as pages of 4 bytes each, below is an example for Ultralight cards with 16 pages:

Card Type - Pages

Card Memory Buffer:

	00	01	02	03
Page 0	00	00	00	00
Page 1	00	00	00	00
Page 2	00	00	00	00
Page 3	00	00	00	00
Page 4	00	00	00	00
Page 5	00	00	00	00
Page 6	00	00	00	00
Page 7	00	00	00	00
Page 8	00	00	00	00
Page 9	00	00	00	00
Page 10	00	00	00	00
Page 11	00	00	00	00
Page 12	00	00	00	00
Page 13	00	00	00	00
Page 14	00	00	00	00
Page 15	00	00	00	00

- Manufacturer Data (Read only)
- Locking bits (one-time programmable)
- User data (one-time programmable)

Lock page 3       Block lock OTP  
 Lock page 4  
 Lock page 5  
 Lock page 6  
 Lock page 7       Block lock 4-9  
 Lock page 8  
 Lock page 9  
 Lock page 10  
 Lock page 11  
 Lock page 12       Block lock 10-15  
 Lock page 13  
 Lock page 14  
 Lock page 15

All pages can be read at once by pressing 'Read all pages from card'. Writing is done one page at a time. To edit a byte simply click on that byte and type in two hexadecimal digits. When ready a page can be written to the card by pressing 'Write to page x'.

Note that Page 3 is one-time programmable which means that once a bit has been set to a 1 it cannot be cleared back to a 0.

To lock any part of the card (which means that the page or block of pages will become read-only) select the relevant 'Lock page x' checkbox and then select page 2 and press 'Write to page 2'.

If you try to write to a locked page you will get a 'Read block error'.

**Note.** NFC tags have dynamic lock bits, the user is advised to refer to the relevant datasheet to prevent inadvertently setting these bits.

# Reading and writing to Mifare card sectors

Select the 'User data blocks' tab and select 'Std 1K', 'Std 4K', 'Plus S', or 'Plus X' card type.

Mifare 1K, 2K and 4K cards are divided into sectors. 1K cards have 16 sectors (numbered 0 to 15), 2K cards have 32 sectors (numbered 0 to 31), and 4K cards have 40 sectors (numbered 0 to 39). To read and write to a card sector you must know the key(s) for that sector. Each sector has a trailer block into which the keys and access conditions for that sector are written. The rest of the sector can be used for any user data.

You will see the following on the left of the screen:

The screenshot shows a software interface for configuring Mifare cards. It features several tabs: 'Keys', 'General', 'Mifare Application Directory', and 'User data blocks'. The 'User data blocks' tab is selected, showing options for 'Secure ID' and 'Sector trailer'. Below the tabs, there are input fields for 'Enter ASCII string to copy into buffer:' and 'Copy string into buffer starting at location: 0x00'. There is also a 'Card Memory Buffer' section with an input field for 'Enter a VALUE to write to a block:' (set to 0) and a checkbox for 'Auto increment value'. A 'Write VALUE to block: 0' button is present. At the bottom, a memory dump table is displayed with columns for hexadecimal values (00-0F) and ASCII characters.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
Blk 0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
Blk 1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
Blk 2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
Blk 3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

And the following on the right of the screen:

Card currently on the programmer:

UID: **05C76CCD**

Working in sector: 1

Current key to access this sector:

Transport key

Mifare card type on programmer:

- Ultralight/NFC
- Std 1K
- Std 4K
- Plus S 2K
- Plus S 4K
- Plus X 2K
- Plus X 4K

Read all blocks from sector 1

Write block 0 to sector 1

Programmer Status

Success.

When you place a blank card on the programmer you will see the card's UID displayed as above. A blank card has the 'Transport' keys already programmed in the trailer block of each sector, so with this key you will be able to read and write to the blocks of the card. The sector in which you wish to work must be set using the updown control next to 'Working in sector'.

Press the 'Read all blocks from sector x' button to read the data from the card in to the memory buffer. Now you can make changes to the memory buffer by:

- Entering a string of text and copying that into the buffer by pressing 'Copy string into ...' Change the position that this string is inserted using the updown control to the right of the button.
- Mifare cards support a VALUE format. Write a 32-bit value into a block using the controls provided. Selecting 'Auto increment value' will result in the value in the box incrementing after every card write.
- Change bytes in hex in the grid.

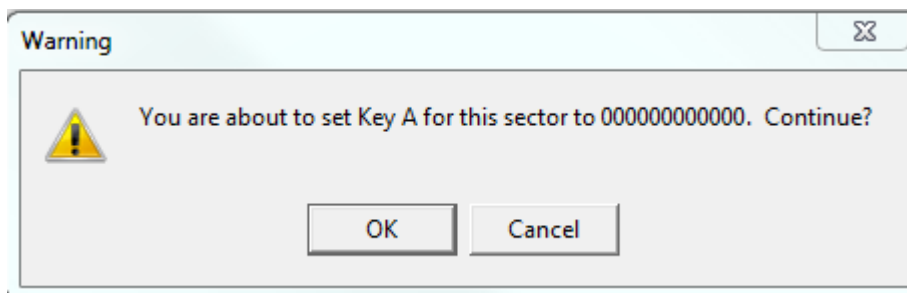
Writing to the card is done one block at a time so select the block you wish to write to by selecting one line in the grid, and then press 'Write block x to sector x'.

Note that you will not be able to write to block 0 of sector 0 as this contains read only manufacturer details (including the UID).

#### **Warning about writing to the last block of each sector**

Note that the last block of a sector (Block 3 for sectors 0-31, and Block 15 for sectors 32-39) contains KeyA, KeyB and the access bits for the sector. It is not advisable to write to this block from the 'User Data Blocks' screen. The main reason for this is explained below:

With Mifare cards it is never possible to read back Key A, so when you do a read of the last block of a sector the first 6 bytes (which is where key A sits) will always read as 0's. If you then write that block back to a card you are in fact setting key A to 000000000000. Thereafter you will have to access this sector using the key 000000000000. To prevent this happening, select 'Cancel' when the software warns you with the following message:



There is a dedicated screen for writing to the last block of a sector. See the section entitled 'Sector Trailer' later.

# Keys

The programmer can store 32 CRYPTO keys and 16 AES keys at any one time. Key 0 should always be left unchanged as the 'Transport' key (FFFFFFFFFFFF). Key 31 is reserved by the program for creating 'Configuration cards' for the 718-xx/724-xx/735-xx/737-xx readers. Keys 1 to 30, and 32 to 47 can be edited by the user.

- **Keys 1-30 may be used for 48-bit CRYPTO keys.**
- **Keys 32-47 may be used for 128-bit AES keys.**

Give each key a meaningful description as once you have entered the key it will be hidden from view and referred to throughout the program by the description:

Index	Description	Type (A or B)	Key (hex)
00	Transport key	A	FFFFFFFFFFFF
01	Canteen application key	A	(hidden)
02	My 718 reconfig key	A	(hidden)
03	MAD read key	A	A0A1A2A3A4A5
04	Secure ID key	A	(hidden)
05	Unused (5)	A	FFFFFFFFFFFF
06	Unused (6)	A	FFFFFFFFFFFF
07	Unused (7)	A	FFFFFFFFFFFF
08	Unused (8)	A	FFFFFFFFFFFF
09	Unused (9)	A	FFFFFFFFFFFF
10	Unused (10)	A	FFFFFFFFFFFF
11	Unused (11)	A	FFFFFFFFFFFF
12	Sector 12 key	A	(hidden)
13	Unused (13)	A	FFFFFFFFFFFF
14	Sector 14 key	A	(hidden)
15	Unused (15)	A	Ff.....

To edit keys, you must fir:

Load from file
Save to file
Show all keys

Once you have entered all the keys you may require for your applications you are advised to save these to a file. **You must also send these keys to the programmer** using the 'Load keys into programmer' button. The keys cannot be read out of the programmer. If you wish to supply the programmer to a third party you should clear the keys in the programmer by pressing 'Clear all keys on programmer'.

The keys that are visible on your screen are not necessarily the same as the keys in the programmer. You must load keys into the programmer.

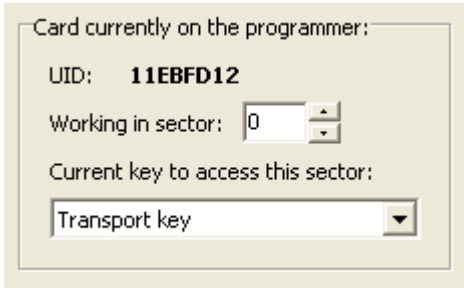
There are two sector keys in each sector trailer block: Key A and Key B. The use of these keys are defined in the access conditions bits. See the 'Sector trailer' tab for all the options. When you enter a key into this program you must define whether the key is to be used as a A or B key. In many applications Key A is used for reading and Key B is used for writing. If you use the same key as both an A key and a B key, then you must enter it twice in the keys screen, once as type A and once as type B.

# Sector Trailer

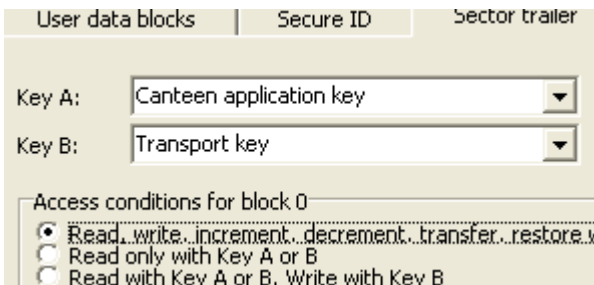
**Note: This screen is dangerous as it is possible to lock out sectors of a card.**

Once you have selected the options to write to the sector trailer of a sector, you will have changed the keys and read/write permissions of that sector. As there are three places on the screen that show keys, it is important to understand how these keys relate.

The key that is shown below is the key that is currently already on the card **before** the operation has taken place. It is the key that is necessary for **writing** to the sector trailer.



The keys shown below are the Key A and Key B that will be written to the card by the operation. They will only exist on the card **after** the operation has been completed.



When experimenting with the program and Mifare cards it is highly recommended that you use the following settings for the access conditions:

**Access conditions for block 0:** Read, write, increment, decrement, transfer, restore with Key A or B.

**Access conditions for block 1:** Read, write, increment, decrement, transfer, restore with Key A or B.

**Access conditions for block 2:** Read, write, increment, decrement, transfer, restore with Key A or B.

**Access conditions for block 3 (sector trailer):** With Key A: can write key A, read and write access bits, read and write Key B.

# Configuration Card for 718-xx readers and 724-xx modules

718-xx/724-xx readers have the following configurable parameters:

- ♦ Mifare card type (Ultralight/NFC tag, Std 1K/4K, Plus 2K/4K).
- ♦ Optional AES authentication for Mifare Plus cards in SL1 mode.
- ♦ Sector Number or Mifare Application ID (when card is configured for MAD).
- ♦ Block Number within the sector (Std 1K/4K) or Page number (Ultralight/NFC)
- ♦ Application Sector Key (must match Key A in the sector trailer for successful authentication)
- ♦ Output format (wiegand, magstripe, RS232)
- ♦ Number of bits to read off the card.
- ♦ First bit position
- ♦ Number of blocks to read
- ♦ Red LED flash - none, on good read, on failed read, on both.
- ♦ Green LED flash - none, on good read, on failed read, on both.
- ♦ Beeper sound - none, on good read, on failed read, on both.
- ♦ UID output - none, on good read, on failed read, on both. (Only available In RS232 mode.)
- ♦ Allow reconfiguration
- ♦ Re-configuration key.

The above parameters are written to a blank 1K Mifare Classic or Mifare Plus S card using the '718-xx/724-xx configuration card' tab.

The screenshot shows a software interface for configuring a Mifare card. It has two tabs: 'Initial Configuration Card' (selected) and 'Re-configuration Card'. The 'Initial Configuration Card' section is divided into several sub-sections:

- Application data to read on user cards:**
  - Mifare card type: Radio buttons for 'Std 1K/4K', 'Ultralight', and 'Plus S / Plus X' (selected).
  - Do AES Authentication: Checked. AES Key: 'SL1 AES Authentication' (dropdown).
  - Application Key: 'Canteen application key' (dropdown).
  - Read operation: Radio buttons for 'Single block (max 16 bytes)' and 'Multiple blocks' (selected).
  - Sector read parameters: 'Number of blocks to read: 3' (spinner), 'Skip sector trailers: Checked'.
  - Application ID (MAD): 'FFFF' (text input).
  - First sector number: '2' (spinner).
- Output:**
  - Format: 'RS232 - 4800 baud' (dropdown).
  - Beep time: '100 ms' (dropdown).
  - Continuous output while card is in the field: Unchecked.
  - Green LED flash: Radio buttons for 'Never', 'On good read' (selected), 'On bad read', 'On any read'.
  - Red LED flash: Radio buttons for 'Never', 'On good read', 'On bad read' (selected), 'On any read'.
  - Beeper sounds: Radio buttons for 'Never', 'On good read', 'On bad read', 'On any read' (selected).
  - UID output: Radio buttons for 'Never' (selected), 'On good read', 'On bad read', 'On any read'.
- Re-configuration card:**
  - Allow reconfiguration in the future: Checked.
  - Choose a new config key: 'My 718 reconfig key' (dropdown).

If you are using the product to create secure ID cards, it may be better to configure your ID cards first using the 'Secure ID' tab. Once you have chosen your ID card format you can use the 'Set Configuration Card settings to match' button to pre-set the values on the Configuration Card screen. Once you have done that you will still have some options to set.

A MIFARE Classic 1k/4k or Mifare Plus S 2k/4k card may be used as a configuration card. The reader uses a factory defined KEY (known to this program as 'Default config key') to read the configuration card. To prevent unauthorised re-configuring of readers in the future, the user can do one of two things:

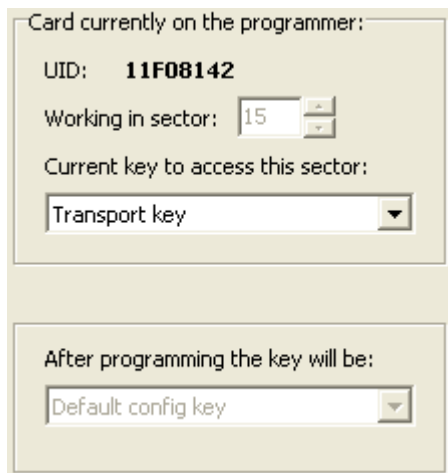
1. Disallow future re-configuration of the reader.
2. Choose a new key to be used for re-configuring the reader.

If you disallow future re-configuration, the reader configuration can only be set back to the factory reset state by reloading the reader firmware. In the above example future re-configuration is allowed using the user defined key 'My 718 reconfig key'.

The application key must match the key that will exist on all the user cards that are presented to the 718-xx reader in normal operation. This key is the one that should be protected to prevent fraudulent use of the cards and reader system. It is therefore recommended that configuration cards are kept in a secure place or destroyed after use (e.g. by overwriting them).

The 'Current key to access this sector' shown below is the key that is currently already on the card **before** the operation has taken place. It is the key that is necessary for **writing** to the sector.

The key that exists in sector 15 **after** the operation, is shown below in the 'After programming the key will be:' box. This is the key that will need to be used to write a new configuration to this card after the first 'Create Configuration card' operation has been done.



The screenshot shows a software interface with two main sections. The top section is titled 'Card currently on the programmer:' and contains the following information: 'UID: 11F08142', 'Working in sector: 15' (with a small spinner control), and 'Current key to access this sector:' followed by a dropdown menu showing 'Transport key'. The bottom section is titled 'After programming the key will be:' and contains a dropdown menu showing 'Default config key'.



# Secure ID

Use this screen to set up a card sector to contain an ID number in a chosen format. This is for use with the 718-xx readers. The user must select the following details of the ID format:

- Electrical output format - Wiegand, Magstripe, or RS232.
- In the case of wiegand:
  - Number of bits e.g. 26.
- In the case of magstripe:
  - Number of characters
  - Number of leading zeros
  - Number of trailing zeros
- In the case of RS232:
  - Number of bytes
- Which block within the sector to write to
- Which bit within the block to start at .

Output format:  
 Wiegand  
 Magstripe  
 RS232

Enter ID number:  
879387  
 Hex  
 Dec  
Load from file ...

Auto increment ID number  
Number of wiegand bits: 26

Card bits (26)  
10000110101101011000110110

Write to block: 0  
First Bit Position: 1  
Fetch settings from Configuration Card  
Set Configuration Card settings to match these

These bytes (hex) will be written to block 0:  
86B58D80000000000000000000000000

If a configuration card has been defined for the reader already, the settings can be used here by pressing the 'Fetch settings from Configuration Card' button.

The programmer will write a bitstream onto the card in the chosen format. The ID number can be entered in Hex or Decimal and can be made to auto-increment after a successful write. The sector trailer needs to be set up first in the 'Sector trailer' screen. It can be programmed either from the sector trailer screen, or from the secure ID screen by ensuring that 'Write to sector trailer when..' is set when the 'Write card bits to block x of sector x' button is pressed.

To load the ID numbers from a text file click 'Load from file...'. The file should contain one ID number per line. After each card write the next ID number in the file will be loaded. When all ID numbers have been read a '0' will loaded into the ID number box.

# Mifare Application Directory

This software will allow you to create a Mifare Application Directory on a Mifare card. Simply enter the application ID that will reside in each sector, define the keys to be used, set the MAD1 or MAD2 format, select the sector to be used by the card publisher, and press 'Write MAD directory to card'. The directory will use sector 0 on a 1K card and sectors 0 and 16 on a 4K card.

The screenshot shows a software window titled "Mifare Application Directory". At the top, there is a "MAD format" section with two radio buttons: "MAD1 - 1K Mifare card" (which is selected) and "MAD2 - 4K Mifare card". Below this is a "Card Publisher's sector:" label followed by a numeric input field containing the value "0". The main part of the window is a table with two columns: "Sector" and "Application ID". The table contains seven rows, with sectors 01 through 07 and all Application IDs set to "0000". To the right of the table are vertical scroll arrows. Below the table are two dropdown menus: "Read Key:" with the selected value "MAD read key", and "Write Key:" with the selected value "My MAD write key". At the bottom of the window are two buttons: "Load from file" and "Save to file".

Sector	Application ID
01	0000
02	0000
03	0000
04	0000
05	0000
06	0000
07	0000

# Configuration Card for 735-xx/737-xx keyboard wedge readers

735-xx/737-xx keyboard wedge readers have the following configurable parameters:

- ◆ Mifare card type (Ultralight/NFC, Std 1K/4K, Plus S/X).
- ◆ Optional AES authentication for Mifare Plus cards in SL1 mode.
- ◆ Sector Number or Mifare Application ID (when card is configured for MAD).
- ◆ Block Number within the sector (Std 1K, Std 4K) or Page number (Ultralight/NFC)
- ◆ Sector Key (must match Key A in the sector trailer for successful authentication)
- ◆ Number of bits to read off the card.
- ◆ First bit position
- ◆ Number of blocks to read
- ◆ Block format (Raw, VALUE, Secure ID wiegand/magstripe)
- ◆ Output format (decimal, hexadecimal, ASCII)
- ◆ Keyboard layout - English, French or International
- ◆ Termination key - none, ENTER, or TAB
- ◆ UID output - none, on good read, on failed read, on both.
- ◆ For VALUE block format:
  - ◆ Number of decimal places
  - ◆ Decimal symbol (dot or comma)
  - ◆ Thousands separator (none, space, dot or comma)

The above parameters are written to a blank mifare card (Mifare Classic 1K/4K, or Mifare Plus S 2K/4K) using the '735-xx/737-xx configuration card' tab.

Output

Format:   Continuous output while card is in the field

Keyboard

English

French

International

Termination

None

ENTER key

TAB key

Display parameters

Hide leading zeros

Number of decimal places:

Decimal symbol

Dot (.)

Comma (,)

Thousands separator

None

Space ( )

Dot (.)

Comma (,)

# Mifare Plus Card Personalisation

Mifare Plus cards are received from the factory in a state called security level 0 (SL0). In order for them to be used in the same way as a Mifare Classic card (Std 1K or Std 4K) they need to be 'personalised'. This involves choosing 5 AES keys which will be written to the card.

## **Card Configuration key**

This key is only used when the card is in security level 3.

## **Card Master key**

This key is only used when the card is in security level 3.

## **Level 2 switch key**

This key is used when switching the security level from 1 to 2.

## **Level 3 switch key**

This key is used when switching the security level from 2 to 3.

## **SL1 AES Authentication key**

This key is used for the optional SL1 Authentication.

Currently the programmer is only able to set the card to security level 1. Only the SL1 AES Authentication key will be used when the card operates at this level. However it is advisable to define valid keys for all of the above, so that in the future the card may be switched to level 2 and level 3 if required. The keys themselves must be entered into the Keys tabsheet into any index position between 32 and 47.

Once you have selected the keys, place the SL0 card onto the programmer. Press 'Check card' to ensure that you have a SL0 card on the programmer. Press 'Personalise Card' to write the keys to the card. Press 'Check card' again to see that the card is now an SL1 card.

# Understanding Mifare Plus

Mifare Plus cards were developed by NXP in response to the cracking of the Mifare Std card (now called Mifare Classic) security algorithm. Mifare Plus cards can be substituted for Mifare Classic cards in many applications and whilst this will be transparent to the end user there are some differences that the application engineer needs to understand.

**Security levels.** Mifare Plus talks a lot about security levels, called SL0,SL1,SL2 and SL3. These describe the different options that the Mifare Plus card can offer the user, briefly these are,

**SL0** - this is the 'blank' card shipping state, similar to the Mifare Classic transport state, except the card is basically useless here, only the UID is readable, the card must be programmed, called 'personalisation', and then set to one of the other higher security levels to be useful to the customer.

**SL1** - in this state the card is equivalent to the Mifare Classic card, but with some of its security flaws fixed. There is one added feature though over the classic card and that is what is called 'optional AES authentication', where a customer can interrogate the card, with their own secret AES key to see if it is a genuine Mifare Plus card issued by them. This is a very nice security feature which prevents card cloning but it does require a Mifare Plus compatible reader. Old style readers will still read the card as before though.

**SL2** - This takes the Mifare Plus card a step higher in security than the old Mifare Classic by using the customers AES keys to communicate and exchange data with the card but still using the block and sector structure of the Mifare classic card. A special card reader is required for these cards.

**SL3** - This is the highest security level where all communication and data exchange use AES keys. It moves away from the NXP Mifare proprietary structure and follows the open ISO14443A -4 specification. A special reader is required for these cards.

In personalising a card the user will choose the required security level to set the card to, a level can only be reached from the previous level, starting at SL0 and the card cannot be returned to a previous lower level.

**UID.** The Mifare Classic card has a 4 byte UID which can be read without any encryption/decryption process. The Mifare Plus card is available in 2 types, type X and type S. The S card has a similar 4 byte UID whereas the X card has a 7 byte UID. For UID only readers the type S card will read and look exactly the same as a Mifare Classic card. For type X cards the UID is read as 7 bytes, similar to a DESFIRE or Ultralight card and should be able to be read, however in the reading process the card identifies itself as a Mifare Plus type X card and some UID only readers which use this information may reject the card as unknown.

**Backward compatibility.** The Mifare Plus S in the SL1 state will work exactly like the Mifare Classic card on any Mifare reader but now includes security fixes for algorithm weaknesses that were exposed on the Mifare Classic card. The Mifare Plus X card is slightly different in that it will look like a Mifare Classic card but only to Mifare readers that have Mifare Plus X compatibility, this is quite important as NXP often promote the Mifare Plus card in SL1 as directly compatible with Mifare Classic, this is only true for Mifare Plus S.