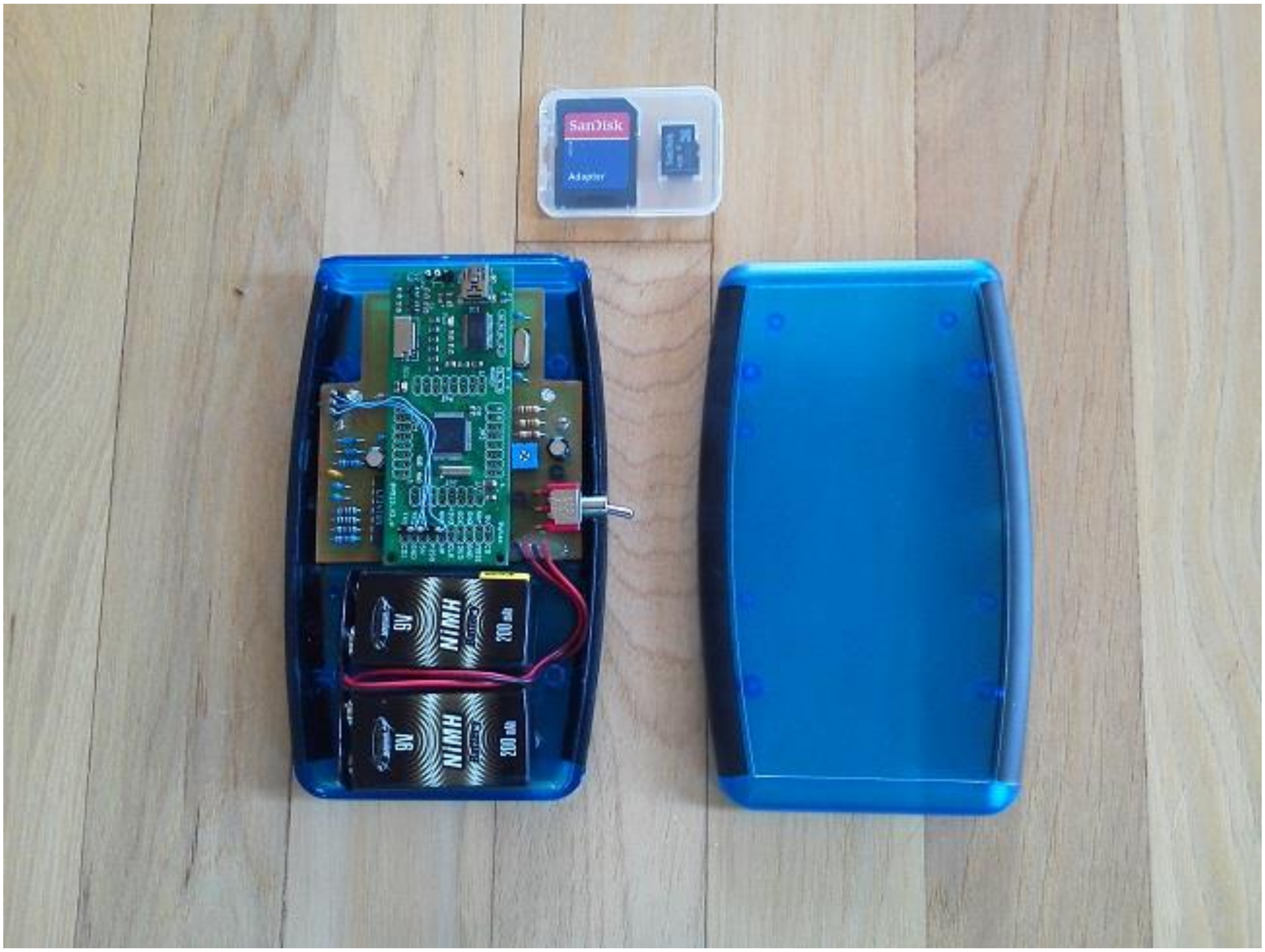

Overview

The High Security Key Extractor unit is used in conjunction with a two part software application program to recover the 128 byte high security key table from an iClass reader that is operating in high security/Elite mode. The unit has been designed to emulate the operation of an iClass card while interacting with an iClass reader during the card/reader mutual authentication sequence.

During the mutual authentication sequence the extractor unit emulates a series of iClass credentials using multiple UID's to obtain the secure message authentication code (MAC) information that is exchanged during the authentication sequence. This MAC information can then be used (offline) to help recover the "High Security/Elite" key information using the brute force attack described in the "Dismantling iClass and iClass Elite" paper found in the following link.

http://www.cs.ru.nl/~flaviog/publications/dismantling.iClass.pdf

The extractor device stores the gathered information to an integral microSD card (text file format) which can then be used offline in conjunction with a two part PC application program to generate the 128 byte high security key table. The unit consists of two microcontroller based circuit boards that are housed in a small handheld blue plastic enclosure. The primary circuit provides the card emulation function while the second circuit manages the microSD card storage of the captured data. The entire unit is powered by two 9Vdc batteries.

The key extraction unit also comes with two separate PC application software programs that are used in conjunction with the microSD text file to brute force the HSkey table in two separate stages. The first stage extracts three bytes (0x00, 0x01, 0x45) of the key table using 2^24 iterations of the MAC Cipher algorithm depicted in the "Dismantling iClass" paper. The approach uses a huge set of lookup tables to obtain the 2^24 Diversified keys needed by the algorithm. The second stage utilizes the same algorithm to recover the remaining bytes of the key table but uses the RW300 reader itself to calculate the Diversified key instead of using a lookup table. Since the recovery of the remaining bytes only requires 2^8 MAC calculations per byte, the diversified key calculation can be done by commanding the RW300 across the RS-232 interface using the iClass Serial Protocol.

Setup

The HS Key Extractor unit requires minimal setup (as described below) prior to being used for the first time.

1. Remove the four screws on the bottom of the blue plastic case.
2. Install the two 9Vdc batteries (provided). Any 9V battery can be substituted if desired.
   [Note: The two batteries provided with the unit are Nickel Metal Hydride rechargeable batteries which can be recharged up to 1000 times using a separate 9Vdc battery charger (not supplied)].
3. Install the microSD card into the microSD card socket on the top circuit board.
   [It should be noted that the SD card contains a file called "hash1.txt" which must always reside on the card. DO NOT delete this file since it provides data used in the creation of the extraction files. A backup copy of the Diversified key Look-up tables are also stored on the card and should not be deleted.]
4. Re-install the four bottom case screws.
   [Note: To maximize battery life and prevent the overheating of the power regulation circuitry, the unit should be turned off whenever the unit is not interacting with an iClass reader. A fully charged set of batteries should provide 1-2 hours of use.]

Operation

The operation of the unit is simple and straightforward as shown below:

1. Power on the device using the toggle switch on the right side of the unit. A green LED on the left side of the enclosure will illuminate to signify that 18Vdc power has been applied.
2. Bring the unit near a HID iclass reader (within 2-3 inches). The antenna used for communication is located on the bottom side of the enclosure. The blue LED located on the right side of the enclosure should begin blinking. It will blink for approximately 15 seconds while the unit is trying to authenticate with the reader using a total of 126 unique Card Serial Numbers (CSN). After 15 seconds the blue LED will be lit solid to indicate that the necessary MAC information has been captured. The LED will remain lit for 3 seconds. If no more sequences are to be captured the reader should be removed from the vicinity of the reader at this time and turned off.
3. If the unit remains near the reader and the three second period has expired the unit will again begin to flash to signify that a new set of extracted data is being captured. This 15/3/15/3 second sequence will be repeated over and over again until the user has captured the desired number of extraction files and removed the unit from the vicinity of the reader.
   [Note: It is recommended that a minimum of two complete sequences be captured from the High Security reader. This is due to the fact that the RF communication sequence will (on occasion) capture a corrupted MAC or nonce value that may prevent the recovery of one or more key table bytes during execution of the offline key recovery program.]

Data captured from the reader that is used to recover the high security key table is stored on the microSD card using unique auto-generated filenames. Since the microcontroller circuit board itself does not contain a real-time-clock (RTC) capability, the filenames used to store the data do not have a valid time stamp associated with them. As a result, filenames are automatically created using a custom algorithm based on remaining storage capacity.   Each time the unit is powered up a unique 4-digit number is created and appended to “mac.txt” to be used as the current filename (e.g. 1234mac.txt). When the unit is held near the reader it will begin capturing one data file every 15 seconds with incrementing filenames similar to the following:
First captured filename: 1234mac1.txt
Second filename: 1234mac2.txt
Third filename: 1234mac3.txt
Etc.

If power is cycled to the unit, the next set of filenames will use an incremented number whose value will be calculated based on the remaining card storage (e.g 1300mac.txt).
As a result, the next series of captured files would be similar to the following:
1300mac1.txt
1300mac2.txt
1300mac3.txt
1300mac4.txt
Etc.

A minimum of one captured file (126 authentication attempts) is required to recover a reader’s high security key table. Subsequent captures are only useful in the event that a corrupted authentication sequence is encountered which will result in the incomplete recovery of the key table. A typical example of a captured set of authentication data (partial) is shown below:

```
Simulated CSN     Hash1(CSN)        Byte  nonce     MAC
000b0ffff7ff12e0  0101000045014545  XX  249d79e2 8090a533
030b0efef7ff12e0  0202000045014545  02  bdb50815 6d83a8c3
040d0dfdf7ff12e0  0303000045014545  03  e350d4df 2ee5f742
040f0ff7f7ff12e0  0901000045014545  09  c604f6d4 6603f649
011310f4f7ff12e0  0C00000045014545  0C  417964ce 3cdfbe32
021410f2f7ff12e0  0E00000045014545  0E  d74077c4 82d0c8e5
051710ecf7ff12e0  1400000045014545  14  7c338407 ed890e2b
006b6fdff7ff12e0  2121000045014545  21  88fb4194 8788b10a
. . .
. . .
Etc.
```

The offline software will iterate all possible key byte values for each hash1 code. It will then use the data fields shown above to calculate a diversified key and a resultant MAC value that will be compared to the expected MAC value captured during the authentication sequence. If a match is made the current key byte value(s) being used for the hash1 code will be assumed to be correct.

Key Table Recovery – Part 1

Once a set of authentication attempts has been captured by the key extractor circuit the file can be used with a separate software application running on a PC to recover the 128-byte high security key table. Part 1 of the key recovery software is used to recover bytes 0x00, 0x01 and 0x45 of the table and Part 2 of the software is used to obtain the remaining 125 bytes.
Recovering the first three bytes of the table is the most difficult and can take up to several days based on the speed of the PC running the application and the value of key byte 0x01. Since the brute force attack must cycle

through a possible 2^24 iterations of the algorithm its execution time will depend heavily on whether key byte 0x01 is a low value (e.g. 0x00) or a high value (e.g. 0xFF)
The following software application must be run to recover the first three bytes of the key table:
key_extract_part1.exe

When this program is run, the user will be prompted to enter the following parameters:

1. The name/location of the key extractor file. A file select window will appear. Use the mouse to select the appropriate directory/file location.
2. The decimal value of the first key value to try (normally always 0). This value corresponds to the value of key 0x01 that will be searched first using a set of Div Key look-up tables (key00.txt through key255.txt)
3. The value of the last key byte value to try (normally always 255).

To verify operation of the program a sample key extraction file has been provided ("test123.txt"). Since key byte 0x01 is known to be 0x3D (61 decimal) the program can be run as shown in the example below to verify operation of the program. If a start value of 61 is entered the recovery should take less than a minute since it starts the brute force attack trying key byte 0x01=61 and because key byte 0x00 is such a low value (0x02).

```
##############################################################
# iClass High Security Key Extractor Utility - Part 1        #
# www.proxclone.com                                          #
# Revision 1.1                                               #
##############################################################

Select Extraction File - Press RETURN to select.
Enter Start of Search Range (0-255):61
Enter End of Search Range (0-255):61
Extraction File Selected: C:\iclass\iclass_cloner_rev7a\extractions\test123.txt
Program started at - 08:42:36

Searching .....
08:42:36 61
08:42:51 - 3 Key Bytes Recovered - Hash1 = 3D3D02027D3D7D7D
Key(0x0) =0x02
Key(0x1) =0x3D
Key(0x45)=0x7D
Search Complete
```

Key Table Recovery – Part 2

Once the first three bytes of the key table have been recovered the second part of the key recovery program can then be run to recover the remaining 125 bytes of the key table. The following application program must be run to complete the key recovery operation.
key_extract_part2.exe

[Note: This program requires that the RW300 reader be attached and powered up. The program uses the RW300 reader to calculate all of the diversified keys that are used in the brute force attack.]

When this program is run, the user will be prompted to enter the following parameters:

1. The serial COM port used to communicate with the RW300 reader (e.g. "4").
2. The values of the three key table bytes obtained from running Part 1 of the key recovery program.
3. The name/location of the key extractor file being used to recover the key table information.
4. The name/location that should be used to store the key table file after it is recovered.

To verify operation of the program a sample key extraction file has been provided (“test123.txt”). Since key bytes 0x00, 0x01, and 0x45 are known (0x02, 0x3D, and 0x7D respectively) the values can be entered into the program to demonstrate (and verify) operation of the program. Each byte of the key table should take less than thirty seconds to recover (approx. 45 minutes total for all bytes). An example of the program operation using this test file is shown below:

```
#############################################################
# iClass High Security Key Extractor Utility - Part 2       #
# www.proxclone.com                                         #
# Revision 1.2                                              #
#############################################################

NOTE: Ensure RW300 Reader is Connected and Powered On

Enter RW300 COM Port No.(1-8): 4
Enter Hex Key(0x00) Value : 02
Enter Hex Key(0x01) Value : 3d
Enter Hex Key(0x45) Value : 7d
Select Extraction File - Press RETURN to select.
Extraction File Selected: C:\iclass\iclass_cloner_rev7a\extractions\test123.txt
Select Key Table Save File - Press RETURN to select.
Key Table Save File: C:\iclass\iclass_cloner_rev7a\hskeytables\test123.key

Program Started at 17:29:03
Searching .....
For MAC = 0x181B2FCB, Recovered Byte 0x02 = 0x4E
For MAC = 0x57667530, Recovered Byte 0x03 = 0x73
For MAC = 0xABE3BE37, Recovered Byte 0x09 = 0x68


……    (output reduced for clarity)

For MAC = 0x40772B78, Recovered Byte 0x05 = 0xCE
For MAC = 0x57F12BF7, Recovered Byte 0x7D = 0xE7
For MAC = 0x32174906, Recovered Byte 0x15 = 0x46
Search Complete at 18:25:37

High Security Key Table

00  02 3D 4E 73 B4 CE 8C C4 83 68 8B 0D 82 82 94 BA
10  0E BD B0 86 C5 46 8E 21 02 D8 E6 66 8D 6B 5F 38
20  71 86 11 FC 90 97 68 E2 7B 11 68 C3 96 1E CB 1F
30  2C 35 EC F2 B4 1D A1 A3 30 B1 CE 18 B0 09 4E 1B
40  E3 EE 7B 40 D6 7D 96 BD 33 EC 6B C7 3B D0 1B 4D
50  EB 3D 06 26 ED BC CA F0 4C 6D 9B 78 46 9A 11 DA
60  D1 A6 A3 1B A9 BB 72 09 C6 E7 09 86 A5 49 83 11
70  29 CF 96 50 90 6E 31 57 C3 BA 5B FD 14 E7 6D 7D

Recovered Key Table Saved to:
C:\iclass\iclass_cloner_rev7a\hskeytables\test123.key
```

Key Table Use

The key table that was recovered from the specified key extraction file is saved to disk in a “*.key” format that is directly compatible with the iClass_Cloner application program. The saved key table can now be imported into the iclass cloner application and used to read and write cards that are compatible with the iclass reader/system where the key extraction operation was performed. See the iClass Cloner instructions for additional details on how the recovered high security key table can be used.

Any problems or questions related to the operation of the High Security Key Extractor software or this set of instructions should be sent to info@proxclone.com .