Product Name	Confidentiality level
mToken CryptoID	
Product version	
V3.0	

mToken CryptoID on Linux/Mac OS

Century Longmai Technology Co., Ltd.

Revision Record

Date	Revision Version	Sec No.	Change Description	Author
2015/12/02	V1.0		Initial Version	Longmai ITD

Product name	Confidentiality level
mToken CryptoID	
Product version	
V3.0	

Contents

1. Introduction

mToken CryptoID is smartcard token compliant with both standards PC/SC and CCID. The mToken CryptoID provides pkcs11 middleware for Linux and Mac OS based on the pcsc-lite. Please make sure the pcsc-lite with CCID driver is installed in your system and add mToken CryptoID support. This document will guide you how to add mToken CryptoID support into Linux and Mac OS.

PC/SC Standard

PC/SC is the de-facto standard to interface Personal Computers with Smart Cards (and smartcard readers of course). Even if PC/SC has been initially promoted by Microsoft -and has been implemented for long time in Windows-, the standard is not limited to MS' operating systems.

PCSC-Lite is an open source implementation of PC/SC, part of a global project named MUSCLE (Movement for the Use of Smart Cards in a Linux Environment). Despite its name, the PC/SC stack offered by MUSCLE is not limited to GNU/Linux anymore. Their compatibility list now includes other popular UNIXes, including Apple Mac OS X and Solaris.

CCID Standard

The USB CCID specification published by the USB Workgroup aims to normalize USB smartcard readers, in order to have a single driver (supplied once for all with the operating system) for virtually any reader from any manufacturer.

2. Support mToken CryptoID on linux

2.1. Install driver on Linux

PCSC-Lite project includes an open source CCID driver. This driver has been tested with mToken CryptoID on GNU/Linux, and should work on the other operating systems targeted by the project.

In this part we're going to install PCSC-Lite and CCID drivers on Ubuntu. Some parts of the procedure may vary a little, depending on the GNU/Linux distribution you're working with.

1. Download

Go to pcsclite.alioth.debian.org to download both PCSC-Lite and its CCID Driver:
PCSC-Lite: file named pcsc-lite-x.y.z.tar.bz2, "x.y.z" being the version number,
CCID driver: file named ccid-x.y.z.tar.bz2, "x.y.z" being the version number.

Switch to root and extract both archives.

Note: On Ubuntu (and on some other Linuxes), connecting as root is not possible. In this case, one must prefix every command by "sudo", to gain root's privileges temporary.

2. Installing PCSC-Lite

Note that on some Linux distributions, the system comes with a pre-installed pcsc-lite.

Before to install pcsc-lite you should also verify that libusb is up to date and if necessary install the dev version.

Open a terminal and go to the directory where you have uncompressed the PCSC-Lite archive. Enter the commands:

```
./configure
make
make install
```

If the configure step fails with message error: usb.h not found, check that you have the libusb library installed. If needed, please install it, and then give the path to this library to the configuration script: ./configure LIBUSB_CFLAGS=....

On Ubuntu or Debian, you may alternatively use apt-get to download and install the libusb-dev package :

```
apt-get install libusb-dev
```

3. Installing CCID driver

Open a terminal and go to the directory where you have uncompressed the CCID Driver archive. Enter the commands:

```
./configure
make
make install
```

2.2. Change Info.plist to support mToken CryptoID

1. Locate the file called Info.plist, for example with a command like this one:

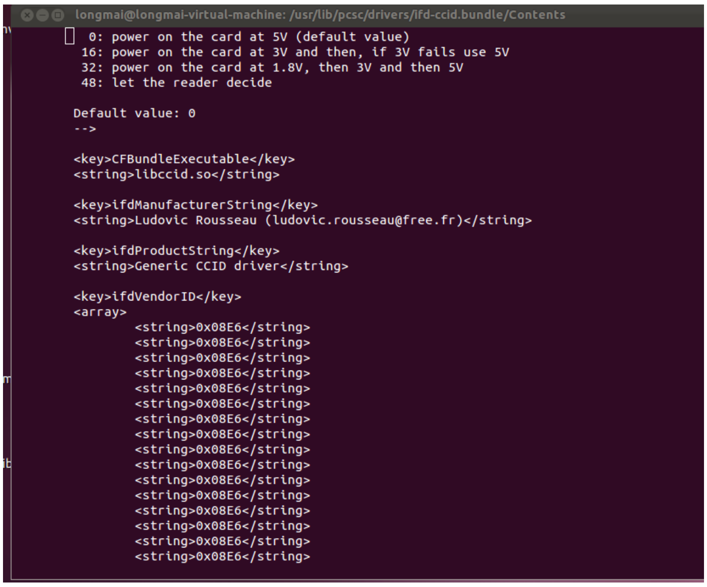
```
sudo find / -name Info.plist
```

You will, for example, find it in
/usr/lib/pcsc/drivers/ifd-ccid.bundle/Contents/Info.plist

Edit Info.plist to add mToken CryptoID.

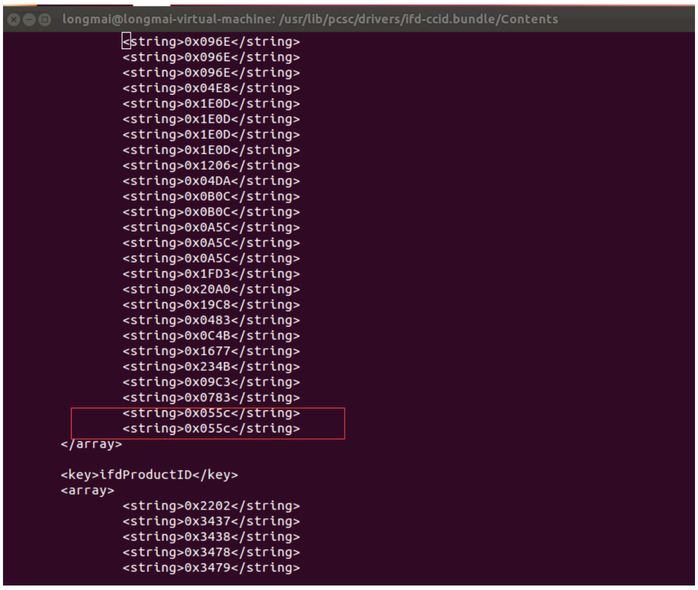
2. Add mToken CryptoID USB Vendor ID

Scroll down and locate the block beginning by <key>ifdVendorID</key>,


longmai@longmai-virtual-machine: /usr/lib/pcsc/drivers/ifd-ccid.bundle/Contents

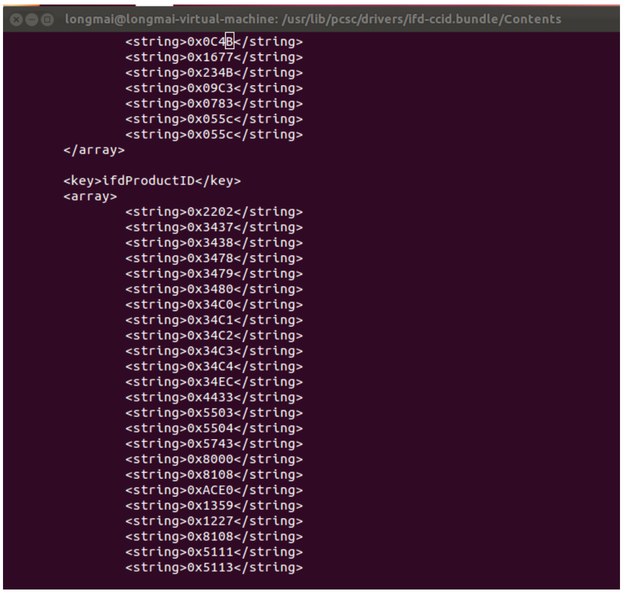
```
          0: power on the card at 5V (default value)
         16: power on the card at 3V and then, if 3V fails use 5V
         32: power on the card at 1.8V, then 3V and then 5V
         48: let the reader decide

        Default value: 0
        -->

        <key>CFBundleExecutable</key>
        <string>libccid.so</string>

        <key>ifdManufacturerString</key>
        <string>Ludovic Rousseau (ludovic.rousseau@free.fr)</string>

        <key>ifdProductString</key>
        <string>Generic CCID driver</string>

        <key>ifdVendorID</key>
        <array>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
                <string>0x08E6</string>
```

Scroll down until you reach the end of the block,
Add 2 occurrences of the line <string>0x055C</string>.


longmai@longmai-virtual-machine: /usr/lib/pcsc/drivers/ifd-ccid.bundle/Contents

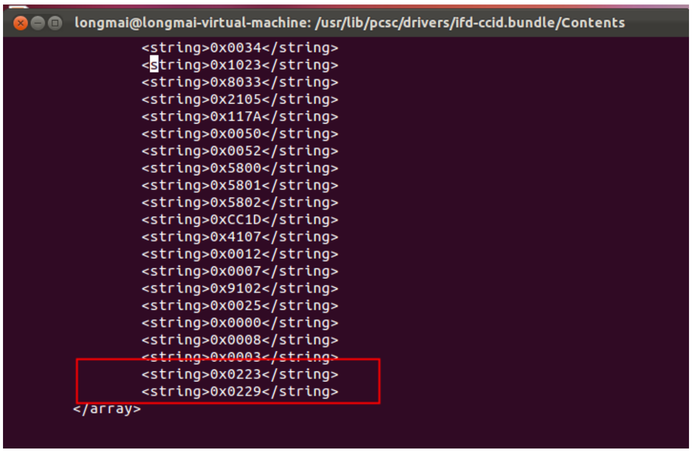
```
                <string>0x096E</string>
                <string>0x096E</string>
                <string>0x096E</string>
                <string>0x04E8</string>
                <string>0x1E0D</string>
                <string>0x1E0D</string>
                <string>0x1E0D</string>
                <string>0x1E0D</string>
                <string>0x1206</string>
                <string>0x04DA</string>
                <string>0x0B0C</string>
                <string>0x0B0C</string>
                <string>0x0A5C</string>
                <string>0x0A5C</string>
                <string>0x0A5C</string>
                <string>0x1FD3</string>
                <string>0x20A0</string>
                <string>0x19C8</string>
                <string>0x0483</string>
                <string>0x0C4B</string>
                <string>0x1677</string>
                <string>0x234B</string>
                <string>0x09C3</string>
                <string>0x0783</string>
                <string>0x055c</string>
                <string>0x055c</string>
        </array>

        <key>ifdProductID</key>
        <array>
                <string>0x2202</string>
                <string>0x3437</string>
                <string>0x3438</string>
                <string>0x3478</string>
                <string>0x3479</string>
```

3. Add mToken CryptoID USB Product ID into ifdProductID part:



```
longmai@longmai-virtual-machine: /usr/lib/pcsc/drivers/ifd-ccid.bundle/Contents
                <string>0x0C4B</string>
                <string>0x1677</string>
                <string>0x234B</string>
                <string>0x09C3</string>
                <string>0x0783</string>
                <string>0x055c</string>
                <string>0x055c</string>
        </array>

        <key>ifdProductID</key>
        <array>
                <string>0x2202</string>
                <string>0x3437</string>
                <string>0x3438</string>
                <string>0x3478</string>
                <string>0x3479</string>
                <string>0x3480</string>
                <string>0x34C0</string>
                <string>0x34C1</string>
                <string>0x34C2</string>
                <string>0x34C3</string>
                <string>0x34C4</string>
                <string>0x34EC</string>
                <string>0x4433</string>
                <string>0x5503</string>
                <string>0x5504</string>
                <string>0x5743</string>
                <string>0x8000</string>
                <string>0x8108</string>
                <string>0xACE0</string>
                <string>0x1359</string>
                <string>0x1227</string>
                <string>0x8108</string>
                <string>0x5111</string>
                <string>0x5113</string>
```

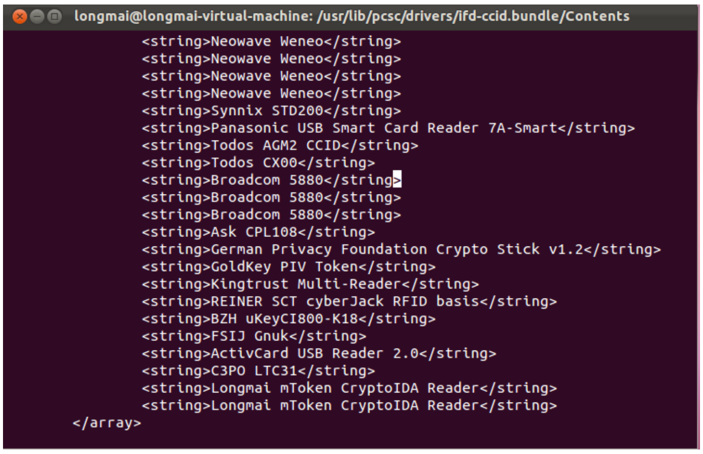
Scroll down until you reach the end of the block, add the following lines:
<string>0x0223</string>
<string>0x0229</string>



```
longmai@longmai-virtual-machine: /usr/lib/pcsc/drivers/ifd-ccid.bundle/Contents
                <string>0x0034</string>
                <string>0x1023</string>
                <string>0x8033</string>
                <string>0x2105</string>
                <string>0x117A</string>
                <string>0x0050</string>
                <string>0x0052</string>
                <string>0x5800</string>
                <string>0x5801</string>
                <string>0x5802</string>
                <string>0xCC1D</string>
                <string>0x4107</string>
                <string>0x0012</string>
                <string>0x0007</string>
                <string>0x9102</string>
                <string>0x0025</string>
                <string>0x0000</string>
                <string>0x0008</string>
                <string>0x0003</string>
                <string>0x0223</string>
                <string>0x0229</string>
        </array>
```

4. Add the names of mToken CryptoID readers

Scroll down, locate the block beginning by <key>ifdFriendlyName</key>, scroll down until you reach the end of the block, and add the following lines:
<string> Longmai mToken CryptoIDA Reader</string>



longmai@longmai-virtual-machine: /usr/lib/pcsc/drivers/ifd-ccid.bundle/Contents

```
                <string>Neowave Weneo</string>
                <string>Neowave Weneo</string>
                <string>Neowave Weneo</string>
                <string>Neowave Weneo</string>
                <string>Synnix STD200</string>
                <string>Panasonic USB Smart Card Reader 7A-Smart</string>
                <string>Todos AGM2 CCID</string>
                <string>Todos CX00</string>
                <string>Broadcom 5880</string>
                <string>Broadcom 5880</string>
                <string>Broadcom 5880</string>
                <string>Ask CPL108</string>
                <string>German Privacy Foundation Crypto Stick v1.2</string>
                <string>GoldKey PIV Token</string>
                <string>Kingtrust Multi-Reader</string>
                <string>REINER SCT cyberJack RFID basis</string>
                <string>BZH uKeyCI800-K18</string>
                <string>FSIJ Gnuk</string>
                <string>ActivCard USB Reader 2.0</string>
                <string>C3PO LTC31</string>
                <string>Longmai mToken CryptoIDA Reader</string>
                <string>Longmai mToken CryptoIDA Reader</string>
        </array>
```

5. Enabling PCSC-Lite daemon

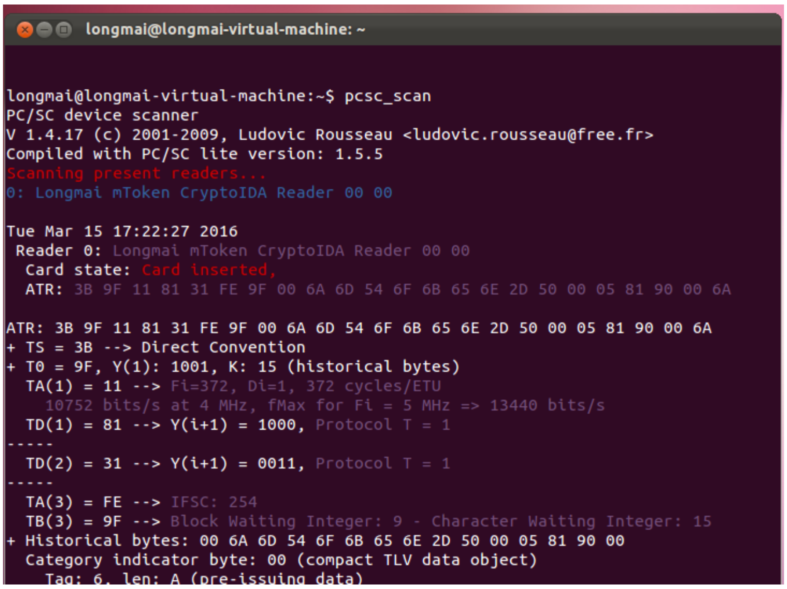
PCSC-Lite's pcscd process must be running in the background. If you have some problems with the daemon you can launch it this way to see some error messages: pcscd -fd
If you see some error messages related to "permission denied", try to launch make fix-rights from the folder where you have downloaded pcsc-lite (use sudo if necessary).

Starting pcscd manually, open a terminal and go to the directory where PCSC-Lite has been installed. Typically, this is /usr/local/sbin/pcscd. In this directory, enter the command. /pcscd, configuring pcscd to be launched on startup.

It is better to have pcscd automatically started when the computer starts. To do so, you must add pcscd in the list of processes started in rc.local or equivalent startup script.

6. Validation

Plug the device onto an unused USB connector, in a terminal, use the command lsusb to verify device's information, if there is usb device with VID 0x055C, the mToken CryptoID is connected normally.
Run pcsc scan to test mToken CryptoID token works normal. From the command line, just type pcsc_scan.



```
longmai@longmai-virtual-machine:~$ pcsc_scan
PC/SC device scanner
V 1.4.17 (c) 2001-2009, Ludovic Rousseau <ludovic.rousseau@free.fr>
Compiled with PC/SC lite version: 1.5.5
Scanning present readers...
0: Longmai mToken CryptoIDA Reader 00 00

Tue Mar 15 17:22:27 2016
 Reader 0: Longmai mToken CryptoIDA Reader 00 00
  Card state: Card inserted,
  ATR: 3B 9F 11 81 31 FE 9F 00 6A 6D 54 6F 6B 65 6E 2D 50 00 05 81 90 00 6A

ATR: 3B 9F 11 81 31 FE 9F 00 6A 6D 54 6F 6B 65 6E 2D 50 00 05 81 90 00 6A
+ TS = 3B --> Direct Convention
+ T0 = 9F, Y(1): 1001, K: 15 (historical bytes)
  TA(1) = 11 --> Fi=372, Di=1, 372 cycles/ETU
    10752 bits/s at 4 MHz, fMax for Fi = 5 MHz => 13440 bits/s
  TD(1) = 81 --> Y(i+1) = 1000, Protocol T = 1
-----
  TD(2) = 31 --> Y(i+1) = 0011, Protocol T = 1
-----
  TA(3) = FE --> IFSC: 254
  TB(3) = 9F --> Block Waiting Integer: 9 - Character Waiting Integer: 15
+ Historical bytes: 00 6A 6D 54 6F 6B 65 6E 2D 50 00 05 81 90 00
  Category indicator byte: 00 (compact TLV data object)
    Tag: 6, len: A (pre-issuing data)
```

2. Support mToken CryptoID on Mac OS

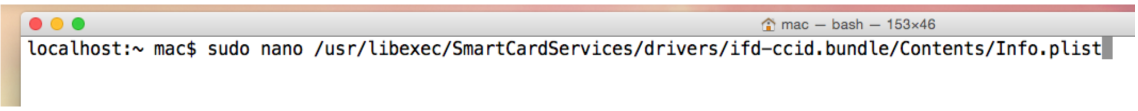
Starting with MacOS X version 10.4, the PCSC-Lite package and the CCID driver are included in the system. We just need to add entries for mToken CryptoID into the configuration file of the CCID driver to have them supported.

2.1 Configuring CCID driver to support mToken CryptoID

1. Open CCID driver's Info.plist

Open a Terminal window and enter this command:

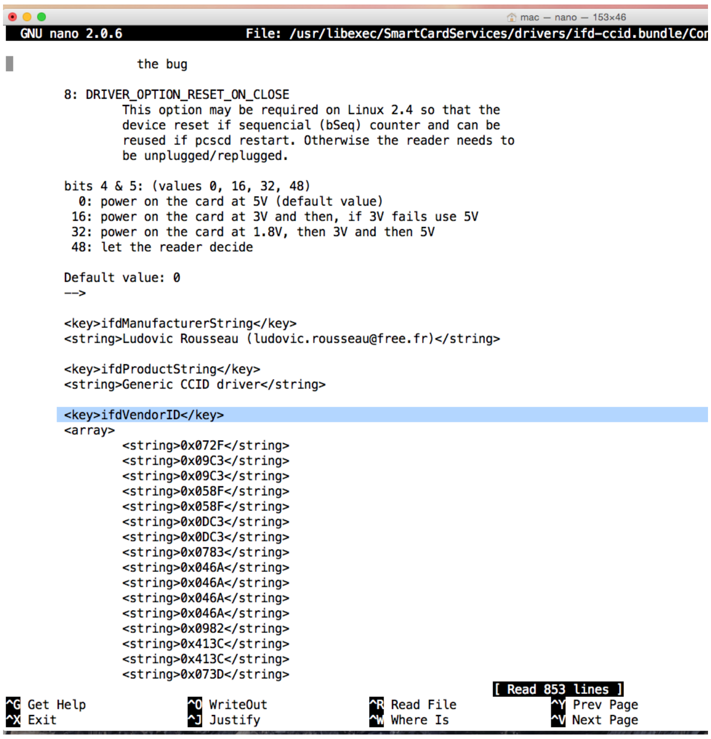
sudo nano /usr/libexec/SmartCardServices/drivers/ifd-ccid.bundle/Contents/Info.plist



```
localhost:~ mac$ sudo nano /usr/libexec/SmartCardServices/drivers/ifd-ccid.bundle/Contents/Info.plist
```

2. Add mToken CryptoID USB Vendor ID

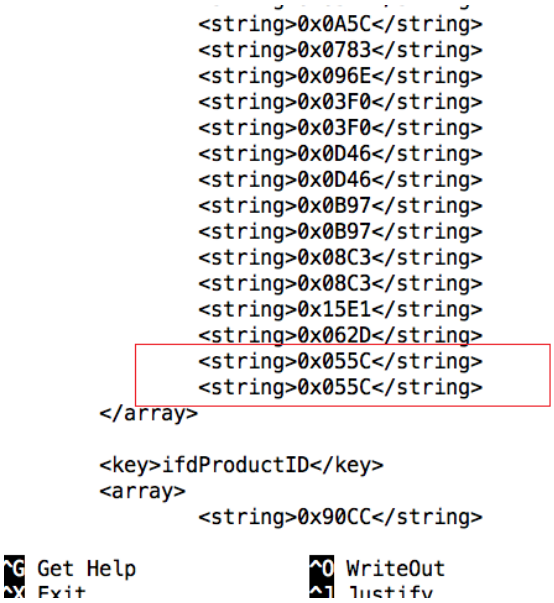
Scroll down, locate the block beginning by <key>ifdVendorID</key>,



```
  GNU nano 2.0.6                 File: /usr/libexec/SmartCardServices/drivers/ifd-ccid.bundle/Co

                  the bug

        8: DRIVER_OPTION_RESET_ON_CLOSE
                This option may be required on Linux 2.4 so that the
                device reset if sequencial (bSeq) counter and can be
                reused if pcscd restart. Otherwise the reader needs to
                be unplugged/replugged.

        bits 4 & 5: (values 0, 16, 32, 48)
          0: power on the card at 5V (default value)
         16: power on the card at 3V and then, if 3V fails use 5V
         32: power on the card at 1.8V, then 3V and then 5V
         48: let the reader decide

        Default value: 0
        -->

        <key>ifdManufacturerString</key>
        <string>Ludovic Rousseau (ludovic.rousseau@free.fr)</string>

        <key>ifdProductString</key>
        <string>Generic CCID driver</string>

        <key>ifdVendorID</key>
        <array>
                <string>0x072F</string>
                <string>0x09C3</string>
                <string>0x09C3</string>
                <string>0x058F</string>
                <string>0x058F</string>
                <string>0x0DC3</string>
                <string>0x0DC3</string>
                <string>0x0783</string>
                <string>0x046A</string>
                <string>0x046A</string>
                <string>0x046A</string>
                <string>0x046A</string>
                <string>0x0982</string>
                <string>0x413C</string>
                <string>0x413C</string>
                <string>0x073D</string>
                                                              [ Read 853 lines ]
^G Get Help             ^O WriteOut             ^R Read File            ^Y Prev Page
^X Exit                 ^J Justify              ^W Where Is             ^V Next Page
```

Scroll down until you reach the end of the block,

Add 2 occurrences of the line <string>0x055C</string>.



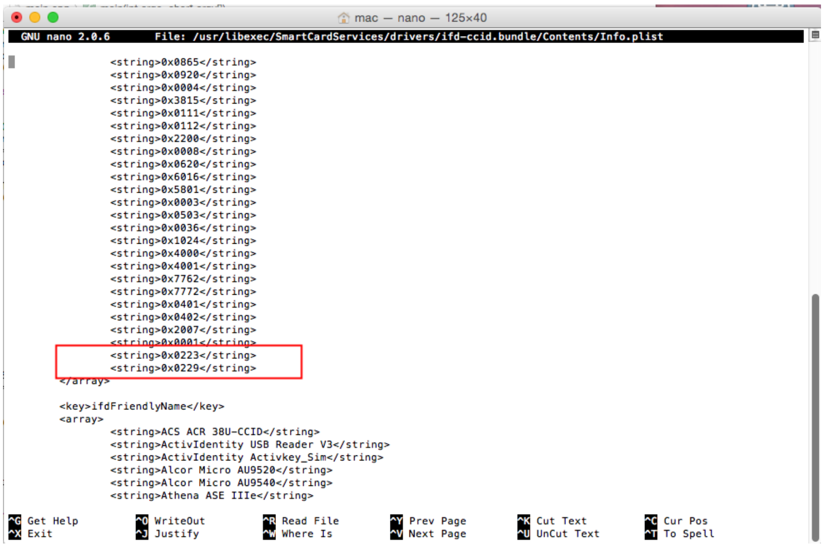
```
        <string>0x0A5C</string>
        <string>0x0783</string>
        <string>0x096E</string>
        <string>0x03F0</string>
        <string>0x03F0</string>
        <string>0x0D46</string>
        <string>0x0D46</string>
        <string>0x0B97</string>
        <string>0x0B97</string>
        <string>0x08C3</string>
        <string>0x08C3</string>
        <string>0x15E1</string>
        <string>0x062D</string>
        <string>0x055C</string>
        <string>0x055C</string>
</array>

<key>ifdProductID</key>
<array>
        <string>0x90CC</string>

^G Get Help            ^O WriteOut
^X Exit                ^J Justify
```

3. Add mToken CryptoID USB Product ID

Scroll down until you reach the end of the block, add the following lines:

<string>0x0223</string>

<string>0x0229</string>

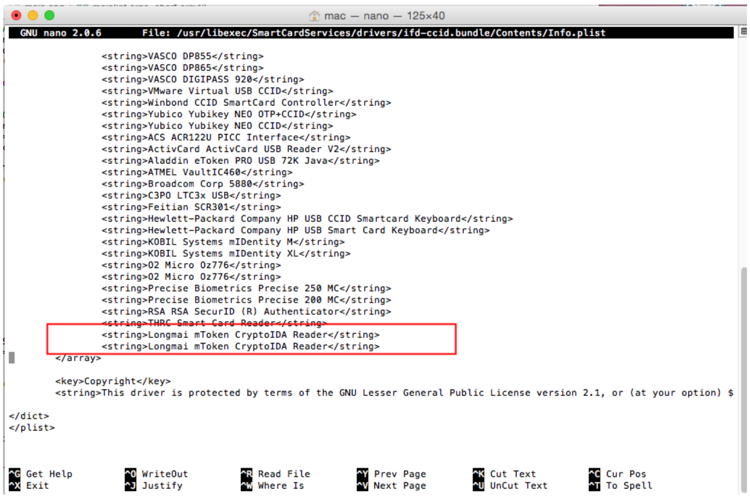


```
  GNU nano 2.0.6        File: /usr/libexec/SmartCardServices/drivers/ifd-ccid.bundle/Contents/Info.plist

                <string>0x0865</string>
                <string>0x0920</string>
                <string>0x0004</string>
                <string>0x3815</string>
                <string>0x0111</string>
                <string>0x0112</string>
                <string>0x2200</string>
                <string>0x0008</string>
                <string>0x0620</string>
                <string>0x6016</string>
                <string>0x5801</string>
                <string>0x0003</string>
                <string>0x0503</string>
                <string>0x0036</string>
                <string>0x1024</string>
                <string>0x4000</string>
                <string>0x4001</string>
                <string>0x7762</string>
                <string>0x7772</string>
                <string>0x0401</string>
                <string>0x0402</string>
                <string>0x2007</string>
                <string>0x0001</string>
                <string>0x0223</string>
                <string>0x0229</string>
        </array>

        <key>ifdFriendlyName</key>
        <array>
                <string>ACS ACR 38U-CCID</string>
                <string>ActivIdentity USB Reader V3</string>
                <string>ActivIdentity Activkey_Sim</string>
                <string>Alcor Micro AU9520</string>
                <string>Alcor Micro AU9540</string>
                <string>Athena ASE IIIe</string>

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

4. Add the names of mToken CryptoID readers

Scroll down, locate the block beginning by <key>ifdFriendlyName</key>, scroll down until you reach the end of the block, and add the following lines:

<string> Longmai mToken CryptoIDA Reader</string>



```
 GNU nano 2.0.6       File: /usr/libexec/SmartCardServices/drivers/ifd-ccid.bundle/Contents/Info.plist

                <string>VASCO DP855</string>
                <string>VASCO DP865</string>
                <string>VASCO DIGIPASS 920</string>
                <string>VMware Virtual USB CCID</string>
                <string>Winbond CCID SmartCard Controller</string>
                <string>Yubico Yubikey NEO OTP+CCID</string>
                <string>Yubico Yubikey NEO CCID</string>
                <string>ACS ACR122U PICC Interface</string>
                <string>ActivCard ActivCard USB Reader V2</string>
                <string>Aladdin eToken PRO USB 72K Java</string>
                <string>ATMEL VaultIC460</string>
                <string>Broadcom Corp 5880</string>
                <string>C3PO LTC3x USB</string>
                <string>Feitian SCR301</string>
                <string>Hewlett-Packard Company HP USB CCID Smartcard Keyboard</string>
                <string>Hewlett-Packard Company HP USB Smart Card Keyboard</string>
                <string>KOBIL Systems mIDentity M</string>
                <string>KOBIL Systems mIDentity XL</string>
                <string>O2 Micro Oz776</string>
                <string>O2 Micro Oz776</string>
                <string>Precise Biometrics Precise 250 MC</string>
                <string>Precise Biometrics Precise 200 MC</string>
                <string>RSA RSA SecurID (R) Authenticator</string>
                <string>THRC Smart Card Reader</string>
                <string>Longmai mToken CryptoIDA Reader</string>
                <string>Longmai mToken CryptoIDA Reader</string>
        </array>

        <key>Copyright</key>
        <string>This driver is protected by terms of the GNU Lesser General Public License version 2.1, or (at your option) $

</dict>
</plist>
```

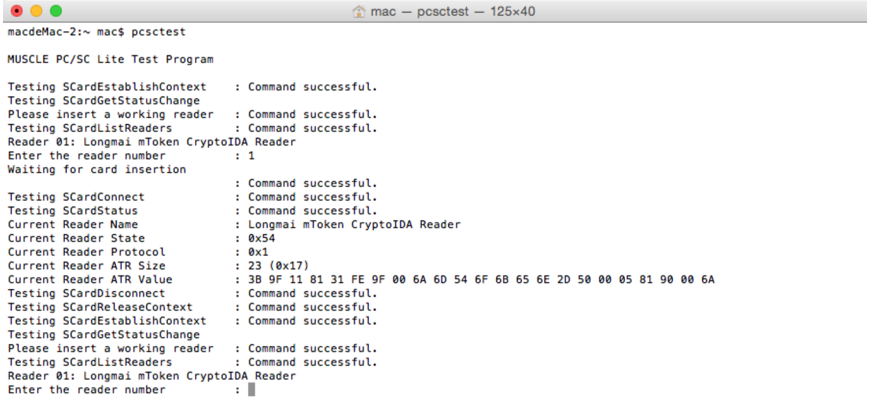
5. Save and restart

Save the modified file. Exit nano and restart the computer.

6. Validation

Plug the reader onto a computer's USB port, Open a Terminal window and enter the command pcsctest, pcsctest displays the list of connected readers *(in the following snapshots, only one mToken CryptoID is connected)*,

Enter the number of the reader you want to test *(in our case, only '1' is allowed)*, and check that everything is running as expected.



```
macdeMac-2:~ mac$ pcsctest

MUSCLE PC/SC Lite Test Program

Testing SCardEstablishContext    : Command successful.
Testing SCardGetStatusChange
Please insert a working reader   : Command successful.
Testing SCardListReaders         : Command successful.
Reader 01: Longmai mToken CryptoIDA Reader
Enter the reader number          : 1
Waiting for card insertion
                                 : Command successful.
Testing SCardConnect             : Command successful.
Testing SCardStatus              : Command successful.
Current Reader Name              : Longmai mToken CryptoIDA Reader
Current Reader State             : 0x54
Current Reader Protocol          : 0x1
Current Reader ATR Size          : 23 (0x17)
Current Reader ATR Value         : 3B 9F 11 81 31 FE 9F 00 6A 6D 54 6F 6B 65 6E 2D 50 00 05 81 90 00 6A
Testing SCardDisconnect          : Command successful.
Testing SCardReleaseContext      : Command successful.
Testing SCardEstablishContext    : Command successful.
Testing SCardGetStatusChange
Please insert a working reader   : Command successful.
Testing SCardListReaders         : Command successful.
Reader 01: Longmai mToken CryptoIDA Reader
Enter the reader number          : 
```

7. In case of trouble

If you are experiencing problem(s), you can, in a terminal, launch this command to see if the daemon reports any error message:

sudo /usr/sbin/pcscd --apdu --debug --foreground

For example you could have a syntax error inside your Info.plist file. You may also check that you have a process called *pcscd*, if that's the case, then the daemon is running.

3. About Century Longmai

Established in 2003, Century Longmai Technology Co., Ltd is one of the most leading information security device vendors in China with over 12 years' experience developing latest generation of digital security solutions and products for secure information access and transmission. Our product portfolios include PKI dongles, wireless PKI tokens, OTP tokens, smart card, smart card readers, electronic document protection solution, software license dongles, Smartcard readers and OEM services. Proved to be secure and convenient, our solutions and products are dedicated to help customers build safe, efficient and sustainable networks, financial systems and enjoy secure access to data and information everywhere whenever they want.

Century Longmai Technology Co., Ltd

3rd Floor, GongKong Building, No.1, WangZhuang Road, Haidian District, Beijing, China
Postcode: 100083
Tel: (86) 10-62323636 | Fax: (86) 10-62313636
Sales E-mail: info@longmai.net
Support E-mail: support@longmai.net
Website: http://www.longmai.net