G+ 更多 下一个博客» 创建博客 登录

Ludovic Rousseau's blog

My activities related to smart card and Free Software (as in free speech).

Friday, September 25, 2015

PCSC sample in Objective-C

To continue the list of PC/SC wrappers initiated in 2010 with "PC/SC sample in different languages" I now present a sample in Objective-C using the Apple Crypto Token Kit API.

Crypto Token Kit API

In Yosemite (Mac OS X 10.10) Apple introduced a new API to access smart cards. See OS X Yosemite and smart cards status.

This API is not a wrapper above PC/SC. It is the native API to be used on Mac OS X. You do not need to install it, it comes with the OS.

Source code

Create a new Cocoa application in Xcode. You need to enable the App Sandbox and add/set the `com.apple.security.smartcard` entitlement to yes.

My sample HellloWorld application does not use Cocoa. It is a text only application.

```
#import <CryptoTokenKit/CryptoTokenKit.h>

int main(int argc, const char * argv[])
{
    TKSmartCardSlotManager * mngr;
    mngr = [TKSmartCardSlotManager defaultManager];

    // Use the first reader/slot found
    NSString *slotName = (NSString *)mngr.slotNames[0];
    NSLog(@"slotName: %@", slotName);

    // connect to the slot
    [mngr getSlotWithName:slotName reply:^(TKSmartCardSlot *slot)
     {
         // connect to the card
         TKSmartCard *card = [slot makeSmartCard];
         if (card)
         {
             // begin a session
             [card beginSessionWithReply:^(BOOL success, NSError *error)
              {
                  if (success)
                  {
                      // send 1st APDU
                      uint8_t aid[] = {0xA0, 0x00, 0x00, 0x00, 0x62, 0x03, 0x01, 0x0C,
 0x06, 0x01};
                      NSData *data = [NSData dataWithBytes: aid length: sizeof aid];
                      [card sendIns:0xA4 p1:0x04 p2:0x00 data:data le:nil
                              reply:^(NSData *replyData, UInt16 sw, NSError *error)
                       {
                           if (error)
                           {
                               NSLog(@"sendIns error: %@", error);
                           }
                           else
                           {
                               NSLog(@"Response: %@ 0x%04X", replyData, sw);
```

Google+ Badge

Ludovic Rousseau blog

G+ Follow

Blog Archive

- ► 2017 (33)
- ► 2016 (49)
- ▼ 2015 (51)
  - ► December (8)
  - ► November (6)
  - ► October (4)
  - ▼ September (3)
    - PCSC sample in Swift
    - PCSC sample in Objective-C
    - Reader Selection: find the smart card reader you s...
  - ► August (5)
  - ► July (1)
  - ► June (4)
  - ► May (3)
  - ► April (3)
  - ► March (2)
  - ► February (5)
  - ► January (7)
- ► 2014 (61)
- ► 2013 (38)
- ► 2012 (27)
- ► 2011 (46)
- ► 2010 (55)

Search This Blog

Search

Subscribe To

Posts

Comments

Google+ Followers

```
                               // send 2nd APDU
                               NSData *data = [NSData dataWithBytes: nil length: 0];
                               [card sendIns:0x00 p1:0x00 p2:0x00 data:data le:@200
                                       reply:^(NSData *replyData, UInt16 sw, NSError
*error)
                                {
                                    if (error)
                                    {
                                        NSLog(@"sendIns error: %@", error);
                                    }
                                    else
                                    {
                                        NSLog(@"Response: %@ 0x%04X", replyData, sw);
                                        NSString *newString = [[NSString alloc] initWi
thData:replyData encoding:NSASCIIStringEncoding];
                                        NSLog(@"%@", newString);
                                    }
                                }];
                           }
                       }];
                  }
                  else
                  {
                      NSLog(@"Session error: %@", error);
                  }
              }];
         } else
         {
             NSLog(@"No card found");
         }
     }];

    // wait for the asynchronous blocks to finish
    sleep(1);

    return 0;
}
```

Output

```
2015-09-25 14:24:19.552 HelloWorld[1578:141676] slotName: Gemalto PC Twin Reader
2015-09-25 14:24:19.668 HelloWorld[1578:141740] Response: <> 0x9000
2015-09-25 14:24:19.681 HelloWorld[1578:141740] Response: <48656c6c 6f20776f 726c6421>
0x9000
2015-09-25 14:24:19.681 HelloWorld[1578:141740] Hello world!
```

Comments

The method `SendIns` is asynchronous. The result is executed in a block. It is similar to a callback in the JavaScript example PCSC sample in JavaScript (Node.js).

With the method `SendIns` you do not specify the class byte. If needed you can use the lower level `transmitRequest` method instead.

The method `SendIns` takes a parameter that contains the data sent to the card. I get a compiler warning if I use `nil` to indicate that I have no data to transmit. I have to create a `NSData` structure of 0 bytes and use it as argument. It is perfectly valid to send no data and the API should allow a simpler code.

My code is a very simple example. The code does not explicitly wait for the asynchronous blocks to finish. I use `sleep(1)` instead. Without this delay the main function would return before the asynchronous blocks are executed.

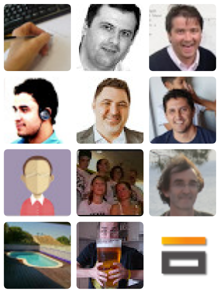
Conclusion

I have seen very few source codes using this new Crypto Token Kit API one year after it is available.
The only API documentation I found is comments contained in the `.h` header files with no sample code.
That does not help.

Maybe the situation will evolve with El Capitan (Mac OS X 10.11) that should be available in the next few days.

Ludovic Rousseau b...

Follow



336 have us in circles

View all

[UPDATE 26 Sept 2015]

It is in fact possible to specify the class byte CLA of an APDU. This byte is stored in the `cla` property of the `TKSmartCard` class. The default value is 0x00.

[UPDATE 31 March 2017]

See also ""PC/SC" sample in Objective-C (synchronous)".

Labels: code, Mac OS X

Newer Post Home Older Post