|  |  |  |  |
| --- | --- | --- | --- |
| Project information | | | |
| Title: | MyID Minidriver Specification | | |
| Status: |  | Revision: | 1 |
| Requested by: |  | Author: | Gerik Rhoden |
| Project ref: |  | Bug ref: | MYIDENT-4869 |

# Introduction

This describes the capabilities of the MyIDVsc smart card minidriver.

This document is targeted at developers who wish to interface with the minidriver directly. This assumes detailed knowledge of the minidriver specification defined by Microsoft.

# Changes since last issue

None.

# Environment

The minidriver only works in concert with the MyID Virtual Smart Card Reader and its associated virtual card applet.

The card has been tested with the “Microsoft Base Smart Card Crypto Provider” and the “Microsoft Smart Card Key Storage Provider”.

# Card types

These cards are supported:

|  |  |  |
| --- | --- | --- |
| Card name | ATR | Key storage |
| Intercede Software Virtual Card | 3B8D0109496E74657263656465120000D4 | Microsoft Software Key Storage Provider (RSA only) |
| Intercede Crypto Virtual Card | 3B8D0109496E74657263656465120002D6 | Any CSP, Microsoft Enhanced RSA and AES Cryptographic Provider |
| Intercede Intel IPT Virtual Card (MyID) | 3B8D0109496E74657263656465120003D7 | Intel Authenticate CSP |
| Intercede Intel IPT Virtual Card (Windows) | 3B8D0109496E74657263656465120004D0 | Intel Authenticate CSP |

# Capabilities

This lists the capabilities and supported properties of the minidriver.

* The minidriver name is MyIDVscDriver.dll
* 32 and 64 bit versions are available
* Versions:
  + Version 6 is the core version.
  + Version 7 function CardCreateContainerEx is supported, as well as version 2 CardRSADecrypt.
  + Version 5 functions CardConstructDHAgreement, CardDeriveKey & CardDestroyDHAgreement are not supported.
* All card operations to the card are atomic. Each function call does just one transaction to card. The minidriver does not use SCard transactions.
* If the card is set to read only, it supports the same set of functions except those banned by the read only requirement.
* Pins:
  + Pin IDs 3 to 7 are supported.
  + The default for PINs 1 and 3 to 7 is 12345678.
  + The default admin PIN (2) is 000000000000000000000000.
  + The IPT card types have modified behaviour for PINs 1 and 3 to 7, see below.
  + All card types support the admin PIN.
  + Only the Intel IPT will display a PIN popup. These cannot be silent.
  + Session PINs are supported where defined by the pin strength property. Session PINS are specific to a Pin ID. Multiple session pins per Pin ID are allowed.
  + Container PIN assignment is supported.
  + Setting the retry count is supported.
  + The optional function CardDeauthenticate is supported.
  + An admin pin (PUK) is not supported.
* Containers:
  + RSA keys may be between 1024 and 16384 subject to key storage limitations
  + ECDSA 256, 384 and 521 are supported where allowed by the key store.
  + There is a maximum of 20 key containers.
* For CardSignData:
  + The cards perform padding. CARD\_PADDING\_PKCS1 is supported, and CARD\_PADDING\_NONE and CARD\_PADDING\_PSS are not be supported by CSP key storage providers.
  + CRYPT\_NOHASHOID is supported. If supplied, the OID is expected to be present in the supplied data.
  + Only the hash algorithms SHA1, SHA256, SHA384 and SHA512 are supported.
  + The CARD\_DATA property pfnCspPadData is not used.
* For CardRSADecrypt:
  + CARD\_RSA\_KEY\_DECRYPT\_INFO\_VERSION\_TWO is supported. Unpadding is done internally.
  + The CARD\_DATA property pfnCspUnpadData is not used.
* Files:
  + The maximum file size is 65519 bytes.
  + Certificate compression is stated as being supported to ensure certificates are sent to the card uncompressed. Neither the minidriver or the card applet read certificates, so compressed certificates may be stored.
  + When the card is created, the file system is fully populated.
* The minidriver does not use the cache functions pfnCspCacheAddFile, pfnCspCacheLookupFile & pfnCspCacheDeleteFile.

# IPT card PIN handling

The two “Intercede Intel IPT Virtual Card” card types change the handling of the PINs as follows:

* Intercede Intel IPT Virtual Card (Windows), user PIN (1)
  + As PINs 3 to 7.
* Intercede Intel IPT Virtual Card (MyID)
  + User PIN (1) is conventional.
  + Containers created for PIN 1 are set to PIN 3.
* PINs 3 to 7
  + These are always shown as logged on, even after a logout.
  + A login always succeeds.

The result of this is that sign and decrypt operations will never prompt for a user PIN, but will always require the use of the Intel Authenticate PIN dialog. The MyID variant has conventional user PIN protection for the purpose of card modification, which provides for extra protection of the card contents.

# Properties

These restrictions on properties apply:

* CP\_CARD\_SERIAL\_NO can be read, but cannot be written.
* CP\_CARD\_AUTHENTICATED\_STATE can be read. Pin ID 0 is always set true.
* CP\_CARD\_CACHE\_MODE only applies to the minidriver session.
* CP\_SUPPORTS\_WIN\_X509\_ENROLLMENT cannot be set.
* CP\_PIN\_CONTEXT\_STRING maximum length is 250 characters.

These additions apply:

* CP\_CARD\_READ\_ONLY can be set, BOOL, requires administrator logon.
* CP\_VSC\_PIN\_POLICY, see below
* CP\_VSC\_DETAILED\_ERROR, see below

The additions are documented in MyIDVscDriverAPI.h:

/// MyID minidriver extra properties

/// This defines extra properties, mostly to do with PIN policy

///

/// Copyright © Intercede Group Plc 2016

#pragma once

/// MyID VSC pin policy card property

/// Permission: Everyone read, admin write

/// \note Used by CardGetProperty() and CardSetProperty()

/// Data type is PinPolicy\_Data of some kind

#define CP\_VSC\_PIN\_POLICY L"VSC Pin Policy"

/// Get the internal error of the last key operation

/// Permission: Everyone read, nobody write

/// \note Used by CardGetProperty()

/// Data type is a DWORD

#define CP\_VSC\_DETAILED\_ERROR L"VSC Detailed Error"

#define VSC\_PIN\_POLICY\_VERSION\_ONE 1

#define VSC\_PIN\_POLICY\_CURRENT\_VERSION VSC\_PIN\_POLICY\_VERSION\_ONE

#define VSC\_MIN\_PIN\_LENGTH 4

#define VSC\_MAX\_PIN\_LENGTH 127

namespace MyID\_Vsc

{

/// Pin policy character option

enum class PinPolicyOption : unsigned long

{

Allow = 0, ///< character allowed, may not be present

Required = 1, ///< character requires at least one

Disallow = 2, ///< character not allowed

};

/// Pin policy data VSC\_PIN\_POLICY\_VERSION\_ONE

/// \note This matches the policy options for Microsoft VSCs on Windows 8.1

class PinPolicy\_Data\_1

{

public:

enum PinPolicySelection

{

UpperCase = 0, ///< A-Z

LowerCase = 1, ///< a-z

Digits = 2, ///< 0-9

Special = 3, ///< special - all printable characters other than letters or digits

///< \note ! " # $ % & ' ( ) \* + , - . / : ; < = > ? @ [ \ ] ^ \_ ` { | } ~

NonPrintable = 4, ///< non-printable - 0x00 - 0x1F and also 0x80 - 0xFF

NumPinPolicy = 5

};

unsigned long m\_version; ///< VSC\_PIN\_POLICY\_CURRENT\_VERSION, set on entry

unsigned long m\_minLength; ///< minimum length, VSC\_MIN\_PIN\_LENGTH - VSC\_MAX\_PIN\_LENGTH

unsigned long m\_maxLength; ///< maximum length, VSC\_MIN\_PIN\_LENGTH - VSC\_MAX\_PIN\_LENGTH

PinPolicyOption m\_policy[NumPinPolicy]; ///< PinPolicySelection

};

typedef PinPolicy\_Data\_1 PinPolicy\_Data;

} // namespace MyID\_VscPin policy

The card supports a PIN policy. This is intended to replicate the policy defined by Microsoft TPM virtual smart cards in Windows 8.1. The property name is CP\_VSC\_PIN\_POLICY and reads and writes object of type PinPolicy\_Data\_1.

## Last error

The last low level error from the key storage provider can be retrieved by getting the property CP\_VSC\_DETAILED\_ERROR, returning a DWORD.

# References

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Title and version** | **Author** | **Date** |
|  | sc-minidriver\_specs\_v7.07.docx | Microsoft |  |
|  | sc-minidriver\_specs\_V6-final.docx | Microsoft |  |
|  |  |  |  |

# Document History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Description of change** | **Section(s)** | **Rev No.** | **Reviewed by** | **Approved by** |
| 6 Mar 2016 | Created |  |  |  |  |
| 22 Aug 2016 | Removed ECC card |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |