

# UNIMATE & UNITOKEN PRO MANUAL

*VERSION 3.0*

***SecuTech***

[www.eSecuTech.com](http://www.eSecuTech.com)

The data and information contained in this document cannot be altered without the express written permission of SecuTech Solution Inc. No part of this document can be reproduced or transmitted for any purpose whatsoever, either by electronic or mechanical means.

The general terms of trade of SecuTech Solution Inc. apply. Diverging agreements must be made in writing.

Copyright © SecuTech Solution Inc. All rights reserved.

WINDOWS is a registered trademark of Microsoft Corporation.

The WINDOWS-logo is a registered trademark <sup>(TM)</sup> of Microsoft Corporation.

## Software License

The software and the enclosed documentation are copyright-protected. By installing the software, you agree to the conditions of the licensing agreement.

## Licensing Agreement

SecuTech Solution Inc. (SecuTech for short) gives the buyer the simple, exclusive and non-transferable licensing right to use the software on one individual computer or networked computer system (LAN). Copying and any other form of reproduction of the software in full or in part as well as mixing and linking it with others is prohibited. The buyer is authorized to make one single copy of the software as backup. SecuTech reserves the right to change or improve the software without notice or to replace it with a new development. SecuTech is not obliged to inform the buyer of changes, improvements or new developments or to make these available to him. A legally binding promise of certain qualities is not given. SecuTech is not responsible for damage unless it is the result of deliberate action or negligence on the part of SecuTech or its aids and assistants. SecuTech accepts no responsibility of any kind for indirect, accompanying or subsequent damage.

## Contact Information

HTTP: [www.eSecuTech.com](http://www.eSecuTech.com)

E-Mail: [Sales@eSecuTech.com](mailto:Sales@eSecuTech.com)

Please Email any comments, suggestions or questions regarding this document or our products to us at: [Sales@eSecuTech.com](mailto:Sales@eSecuTech.com)

Version	Date

### CE Attestation of Conformity



UniToken is in conformity with the protection requirements of CE Directives 89/336/EEC Amending Directive 92/31/EEC. UniToken satisfies the limits and verifying methods: EN55022/CISPR 22 Class B, EN55024:1998.

### FCC Standard



This device is in conformance with Part 15 of the FCC Rules and Regulation for Information Technology Equipment.

Operation of this product is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.



The equipment of UniToken is USB based.

### Conformity to ISO 9001:2000



The Quality System of SecuTech Solution Inc., including its implementation, meets the requirements of the standard ISO 9001:2000

### ROHS



All UniMate & UniToken products are environmental friendly with ROHS certificates.

# Table of Contents

PART 1 AN OVERVIEW OF UNIMATE & UNITOKEN	2
CHAPTER 1: UNIMATE & UNITOKEN DEVICE	2
1.1 Features	2
1.2 Specifications	3
CHAPTER 2: UNIMATE & UNITOKEN SOFTWARE	3
2.1 UniMate & UniToken driver installation	3
2.2 The PKCS#11 and MS-CAPI Modules of UniToken	4
2.3 Token API	4
2.4 Supported Platforms	4
CHAPTER 3: SECURITY	5
3.1 Key	5
3.2 Data transmission	7
3.3 Factory Default Settings	7
PART 2 UNIMATE & UNITOKEN SDK	7
CHAPTER 4: SDK OVERVIEW	7
4.1 Driver installation	8
4.2 Redistribution Package	8
4.3 Console	12
4.4 Monitor	42
PART 3 APPLYING DIGITAL CERTIFICATES	55
CHAPTER 1: APPLYING DIGITAL CERTIFICATES	55
1.1 Applying VeriSign Certificates	55
1.2 Applying Microsoft Certificates	56
1.3 Using Digital Certificates	58
PART 4 DEVELOPER'S GUIDE	59
1.1 Device Initialization	59
CHAPTER 1: PKCS11 APPLICATION	59
1.2 Introduction	59
1.3 Supported PKCS#11 Algorithms and APIs	61
1.4 UniMate & UniToken PKCS#11 Function Library	62
1.5 Samples	65

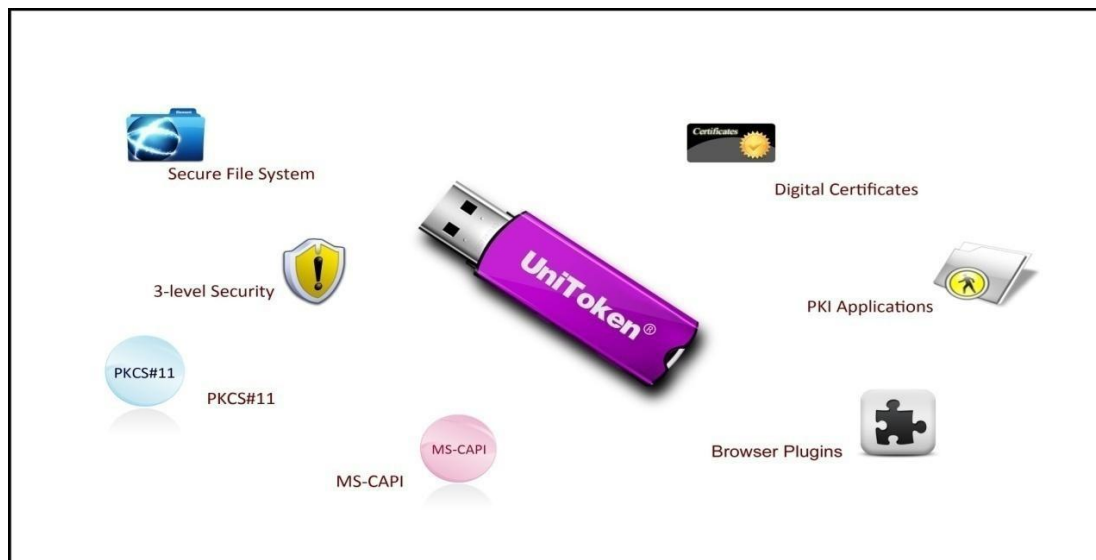
CHAPTER 2: MS-CAPI APPLICATIONS	67
2.1 Introduction	67
2.2 Supported Algorithms and APIs	68
2.3 Samples	69
2.4 UniMate & UniToken API	72

# Part 1 An Overview of UniMate & UniToken

UniMate & UniToken, hereinafter referred to as Token, is an information security product based on CCID technology. It is a secure container for digital credentials. Advanced processor and secure memory are built in the Token device to guarantee the security for exchanging, storing and handling electronic information.

Token has achieved an effective rights management and can provide a highly-secured file system. A built-in computing engine accomplishes fast and efficient information processing.

Token supports PKI applications and provides Token API for secondary development. Abundant samples bring ease to integrations.



## Chapter 1: UniMate & UniToken Device

### 1.1 Features

Key features of UniMate & UniToken device:

- Globally unique hardware ID
- Customized software ID
- Smartcard-based

- On-board encryption
- Two levels of PIN management mechanism
- A secure file system
- Large memory –up to 64K
- Stylish and cute case
- Lead free

## 1.2 Specifications

Dimensions	57×16×8 mm
Weight	9g
Min. Operating Voltage	5V
Current Consumption	<= 50 mA
Operation Temperature	0°C to 70°C
Storage Temperature	-10°C to 85°C
Humidity Rate	0-70% without condensation
Casing	Tamper-evident Metal
Memory Data Retention	At least 10 years
Memory Cell Rewriters	At least 100,000 times

# Chapter 2: UniMate & UniToken Software

## 2.1 UniMate & UniToken driver installation



## 2.2 The PKCS#11 and MS-CAPI Modules of UniToken

PKCS#11 module of Token is implemented according to PKCS#11 standards V.2.20, which is a DLL file for C language running on Windows operating system. MS-CAPI Module of UniToken is implemented in line with MS-CAPI standard.

These two modules can be used in cooperation with each other, i.e. the certificate applied with PKCS#11 can be used by MS-CAPI module of Token, and vice versa.

## 2.3 Token API

Token provides a set of Token API, which allows users to manage one or several Token hardware keys, i.e. operation of Token attributes, permission, built-in algorithms and secure file system. Please install Token API package or Token full package to enable these features.

## 2.4 Supported Platforms

Table 1.3: Supported Platforms

Components	UniMate Flex	UniMate STD	UniToken PRO
OS			
Windows 2000	√	√	√
Windows 2003	√	√	√
Windows XP	√	√	√
Windows Vista	√	√	√
Windows 7	√	√	√
Windows 2008	√	√	√
Windows 2012	√	√	√
Windows 8	√	√	√

iOS	√		
Android	√		

## Chapter 3: Security

Security is the most important part in Token system, which involves in identification and verification method, including not only the file access permission control mechanism inside the token, but also the information confidential control inside the token. The security attribute means the current state of the device when the card is reset or after the token finished some commands.

### 3.1 Key

The following table describes different key types and use

Key Type	Use
Transmission Key	Ensure the security during the card initialization, and provide encryption and decryption.
PIN	Directory level authentication. control different users' read and write permission
PIN unlock key	Used to unlock PIN
PIN reload key	used to reload PIN
External authentication key	Token uses this key to authenticate the external entity
Internal authentication key	External entity uses this key to authenticate the token device.
Master key	Used to secure transmission
Block encryption/decryption key	Provide encryption/decryption for external entity.

Transmission key: a 16-byte key that every device must have only one transmission key

PIN: a personal identification number based on directory. The PIN is firstly hashed and then stored in the device

PIN unlock key: a 16-byte key is used in unlock function. Its function is that encrypts PIN and calculates MAC of the cipher text as a key.

PIN reload key: not used in this version and will add this function in the following version.

External authentication key: a 16-byte key that used for external authentication. The first 8-byte is the key1 and key3.

Internal authentication key: a 16-byte key that used for internal authentication. The first 8-byte is the key1 and key3.

The block encryption/decryption key: used to specified algorithm, length is from 8-byte to 16-byte. Currently the supported algorithms are DES (ECB, CBC), TDES (ECB, CBC), AES (EBC, CBC).

Authentication type	Key type	Use method and algorithm
Access permission authentication	Transmission key	Comparison in plaintext  External authentication (for example, format device in user state)
	PIN	External authentication (TDES)
	Extern authentication key	External authentication (TDES)
	Internal authentication key	Internal authentication (TDES)
Cipher text transmission	Transmission key	TDES encryption (use DES in MAC)
	Master key	TDES encryption (use DES in MAC)
	PIN unlock key	TDES encryption (use DES in MAC)
	PIN reload key	TDES encryption (use DES in MAC)
Provide encryption operation for external entity	Encryption key	Depending on implemented encryption algorithms.

## 3.2 Data transmission

Data transmission means data transmitted between host machine and device, including 4 transmission modes.

Mode	Definition	Security	Integrity
Plaintext	Data is transferred directly without any process	×	×
Plaintext with MAC	Plaintext and MAC of the plaintext are transferred together	×	✓
Cipher	Plaintext is encrypted before transferred	✓	×
Cipher with MAC	Data is encrypted and calculate the MAC of the encrypted data, and then transferred the cipher text and MAC	✓	✓

## 3.3 Factory Default Settings

The default issue transmission key is "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF" and the default master key is "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF". Users shall use the master key to create and delete file.

# Part 2 UniMate & UniToken SDK

## Chapter 4: SDK Overview

Table1.6 Token SDK Contents

Components	Description
Include	Declaration of the standardized identifiers and interface of PKCS #11, CSP and Token API.
Libraries	Token libraries
Documents	Manual for Token PRO and API

	reference
Integration Guides	Instruction about integration Token with other software
Redists	Redistribution packages for developers and end users
Samples	Samples for CSP, PKCS and Token API
Windows CCID Driver	Token Drivers

## 4.1 Driver installation

Most of the latest Windows systems (Windows 7 and later) don't need install any driver to make Token work. For some old versions, such like Windows VISTA and XP, driver must be installed to make the system recognise the device.

After inserting Token to a computer, from Control Panel → Hardware and Sound → Device Manager, open the Device manager.

From the hardware list find the unknown device, update the driver, the driver is in the SDK\windows CCID Driver.

## 4.2 Redistribution Package

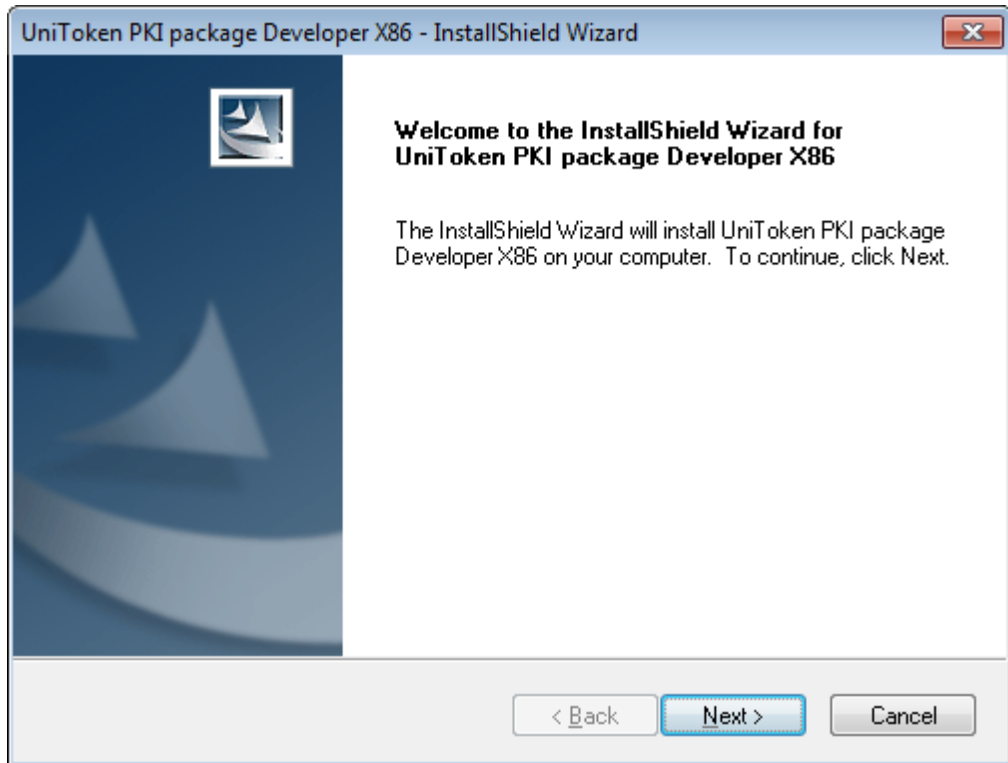
Token provides two different redistribution packages for developers and end users respectively. Both the package provide Token PKI installation package. If you want to use the PKI application, you must install it.

- Installation

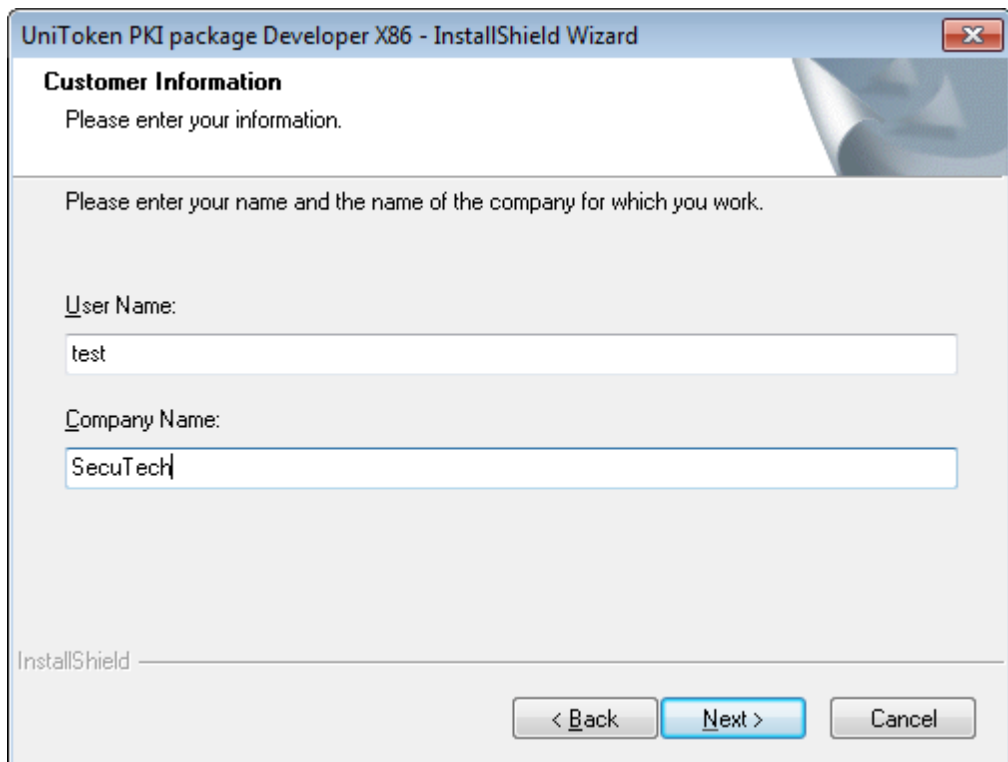
Token PKI package can be found in the redist folder of Token PRO SDK.

For developers package

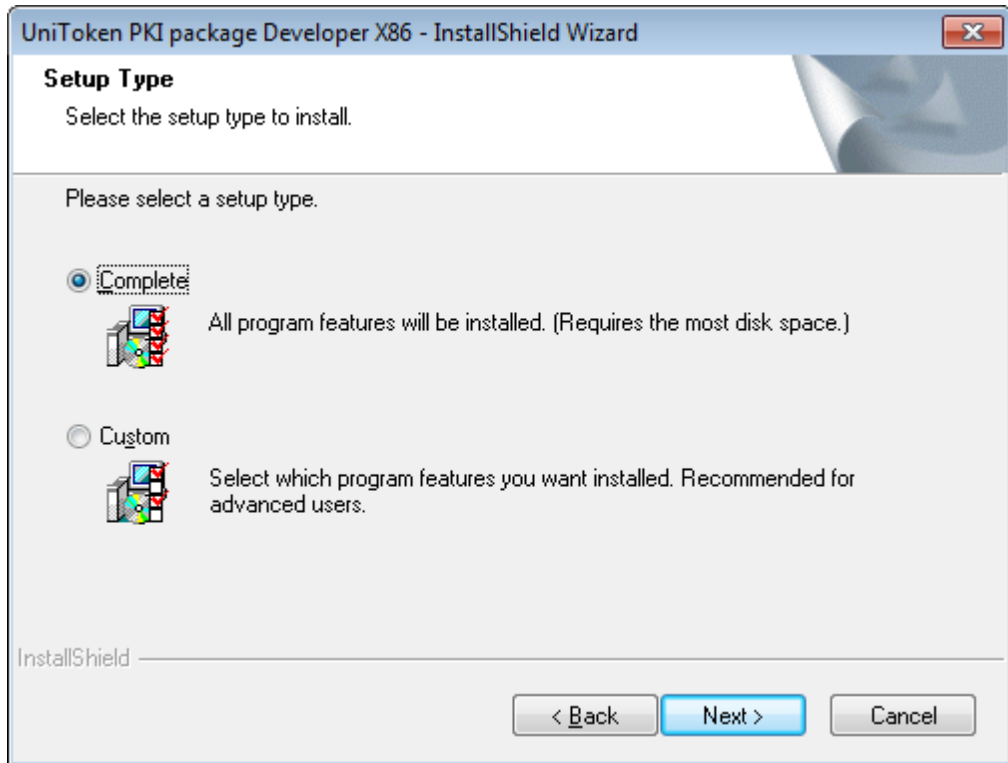
Double click the icon to run the install shield wizard, and follow the illustration below:



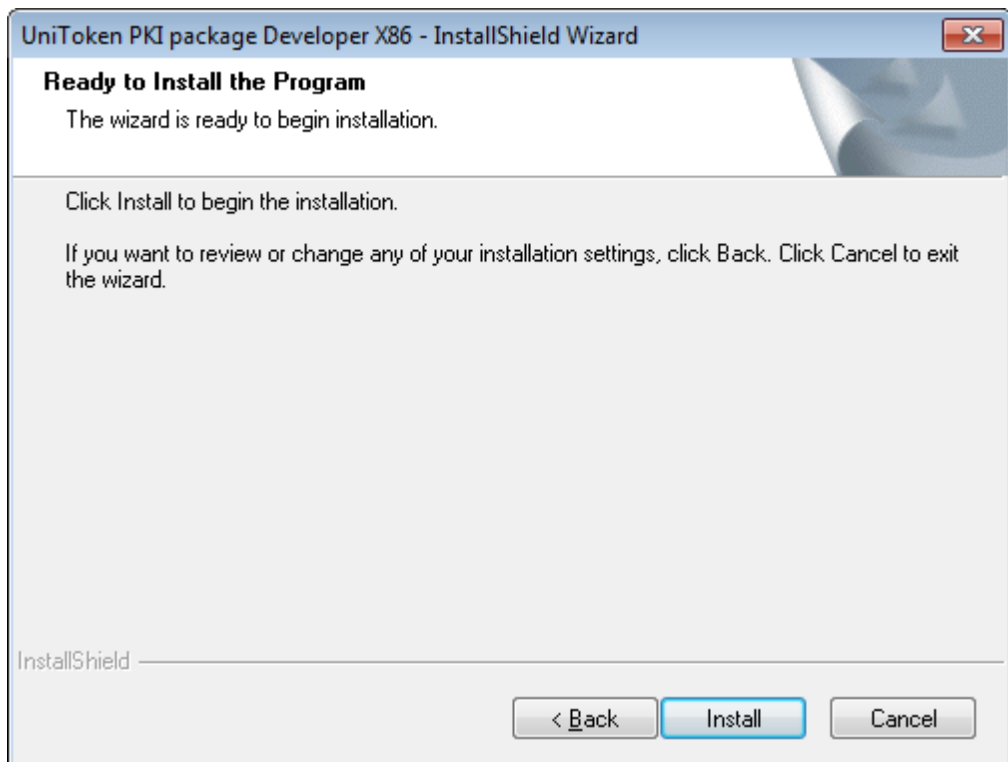
Click "Next".



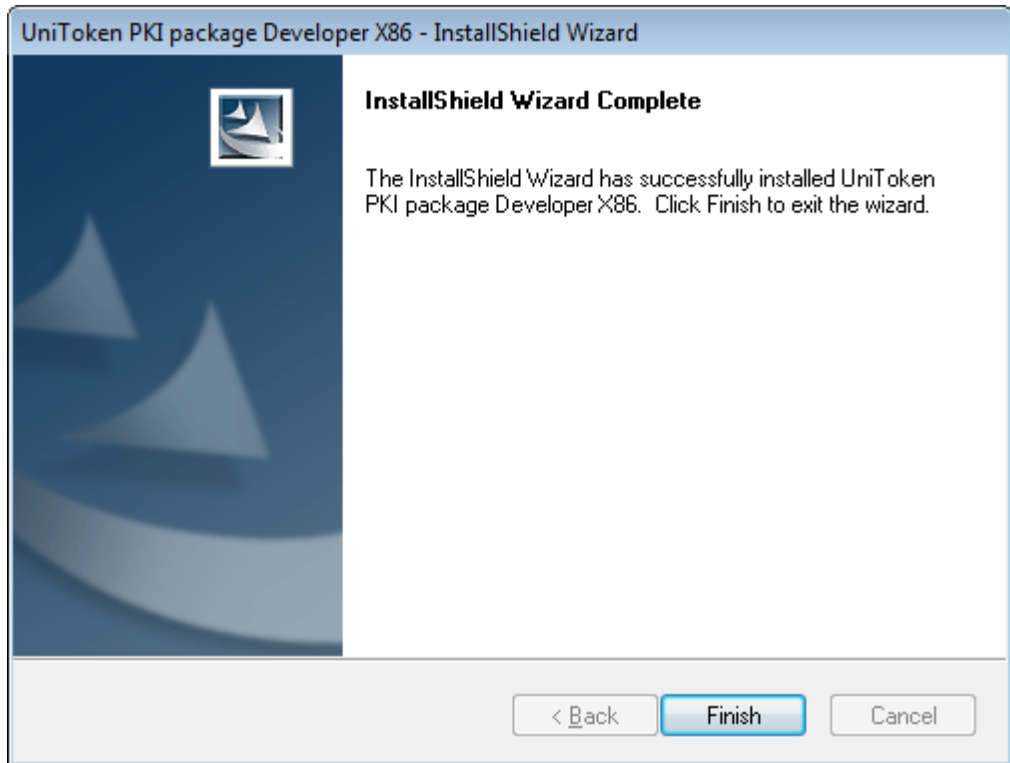
In this section, user name and company name are required. And click "Next".



Users are allowed to choose setup type. And click "Next".



Click "Install".



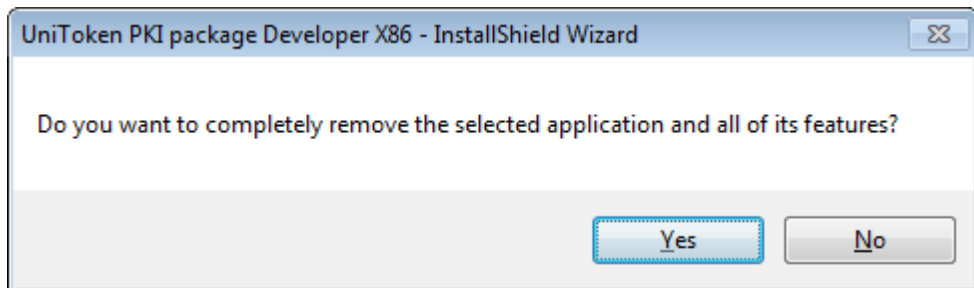
At last, click “Finish” to close the installing wizard.

- Uninstallation

To uninstall the software, there are two ways: start menu and control panel.

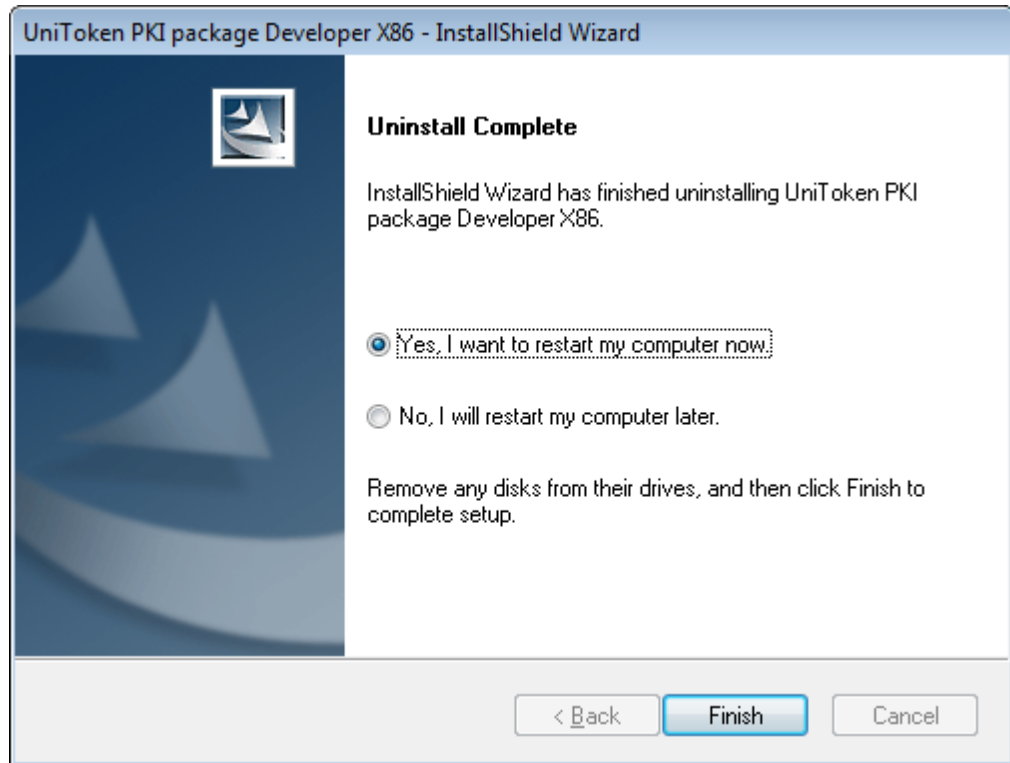
Start Menu:

Select “Start-All Programs-SecuTech-Token-Uninstall Token PKI package”



Click “Yes”.





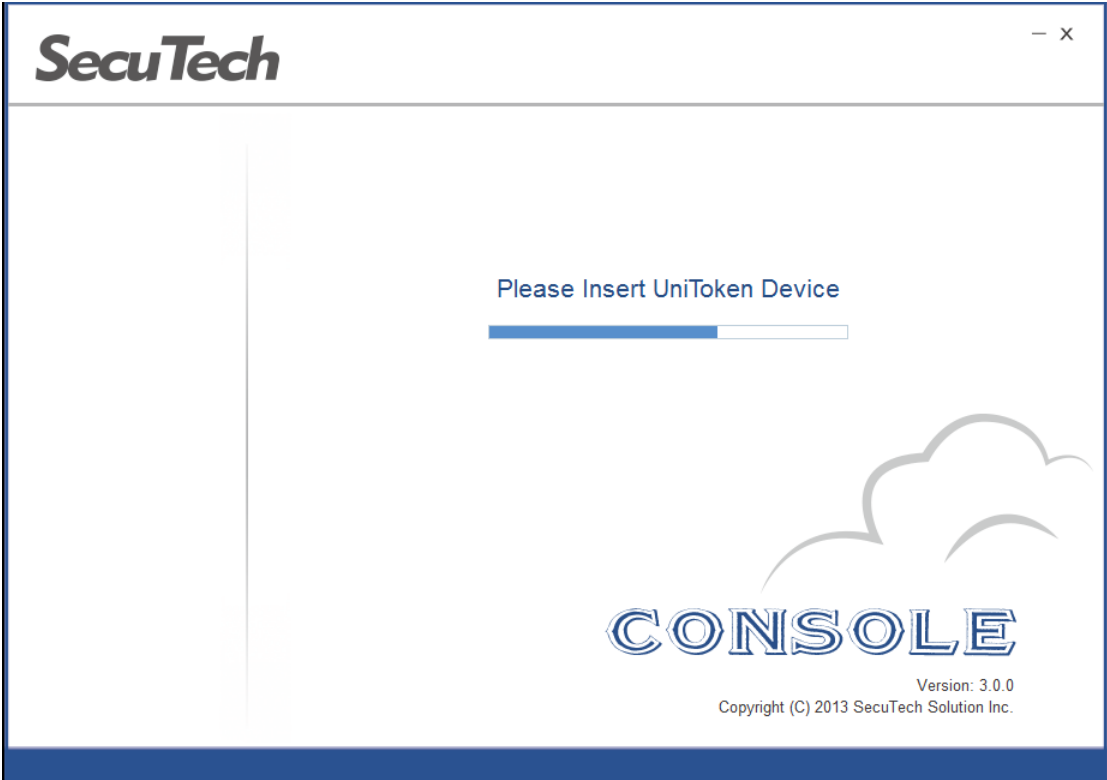
Click "Finish".

## 4.3 Console

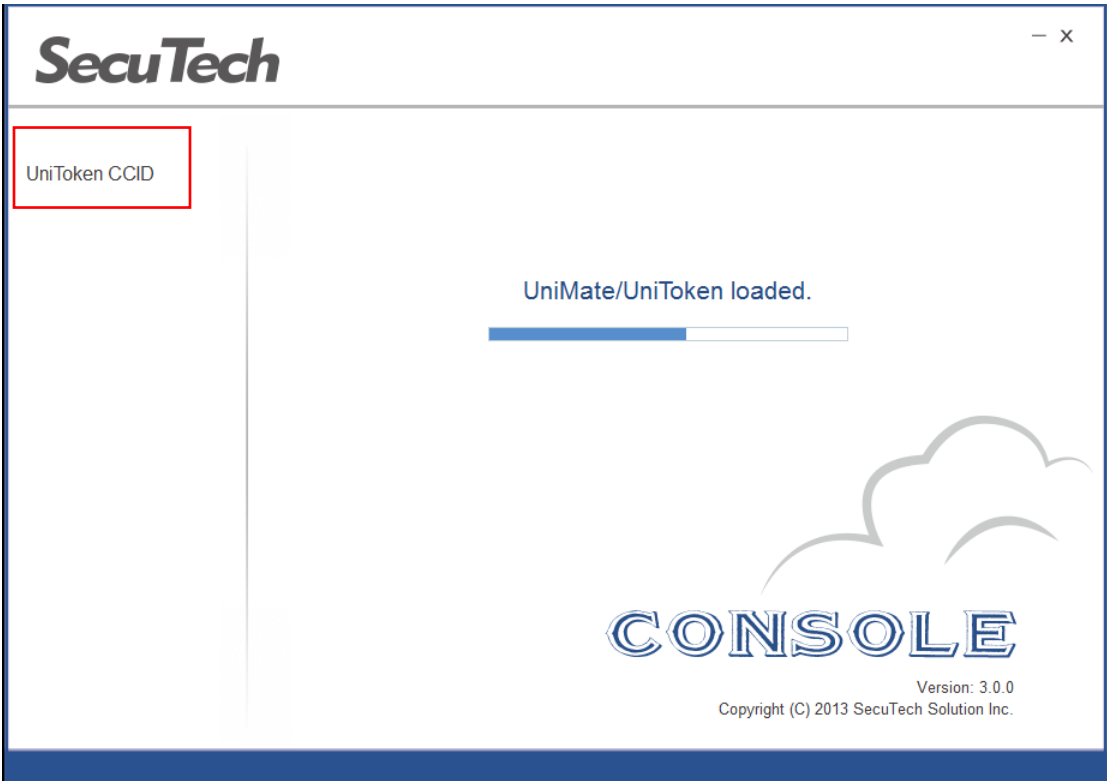
Token Console is used to manage devices, set user permission as well as control file system and certificates.

### 4.3.1 *Check Token information*

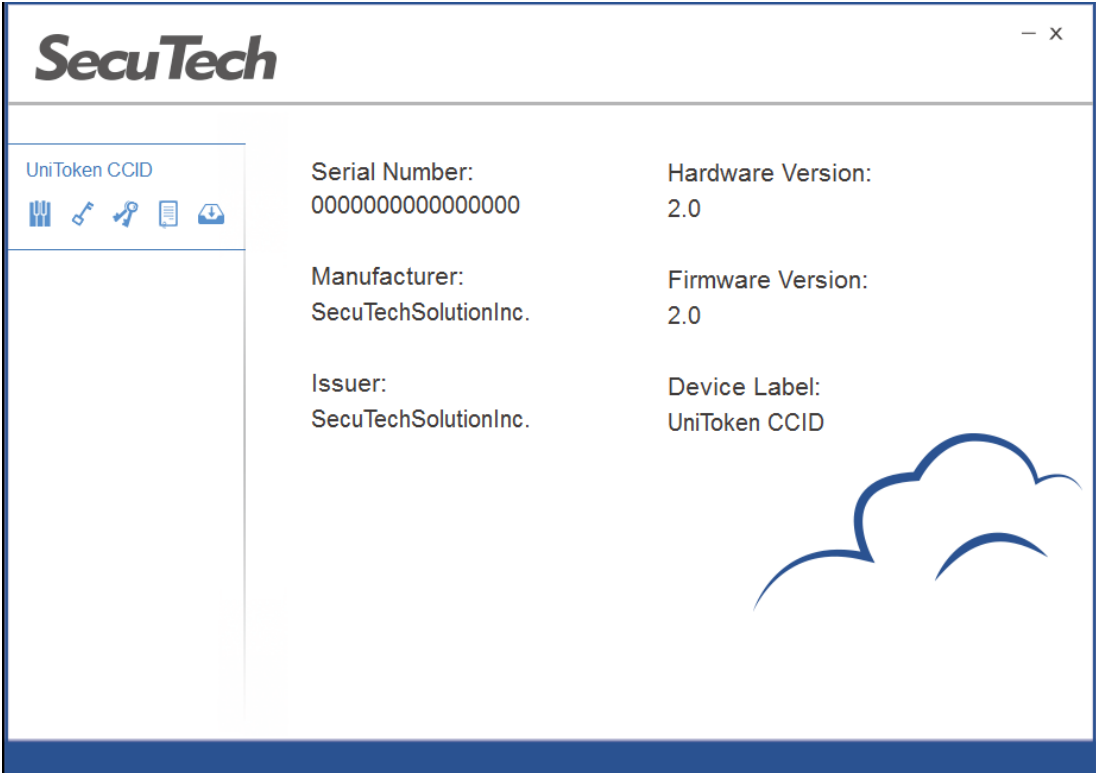
1. Start Console.exe and insert your device



The device name will appear on the left side of the page.

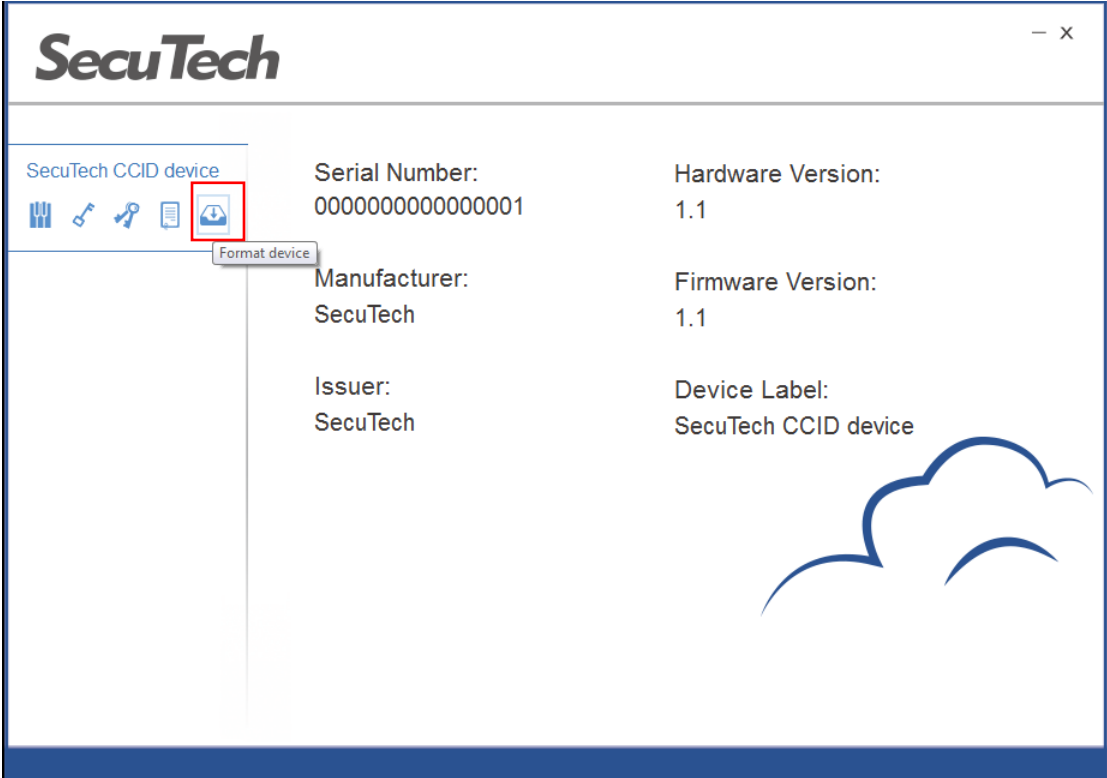


2. Click on the name of the device to check the device information.

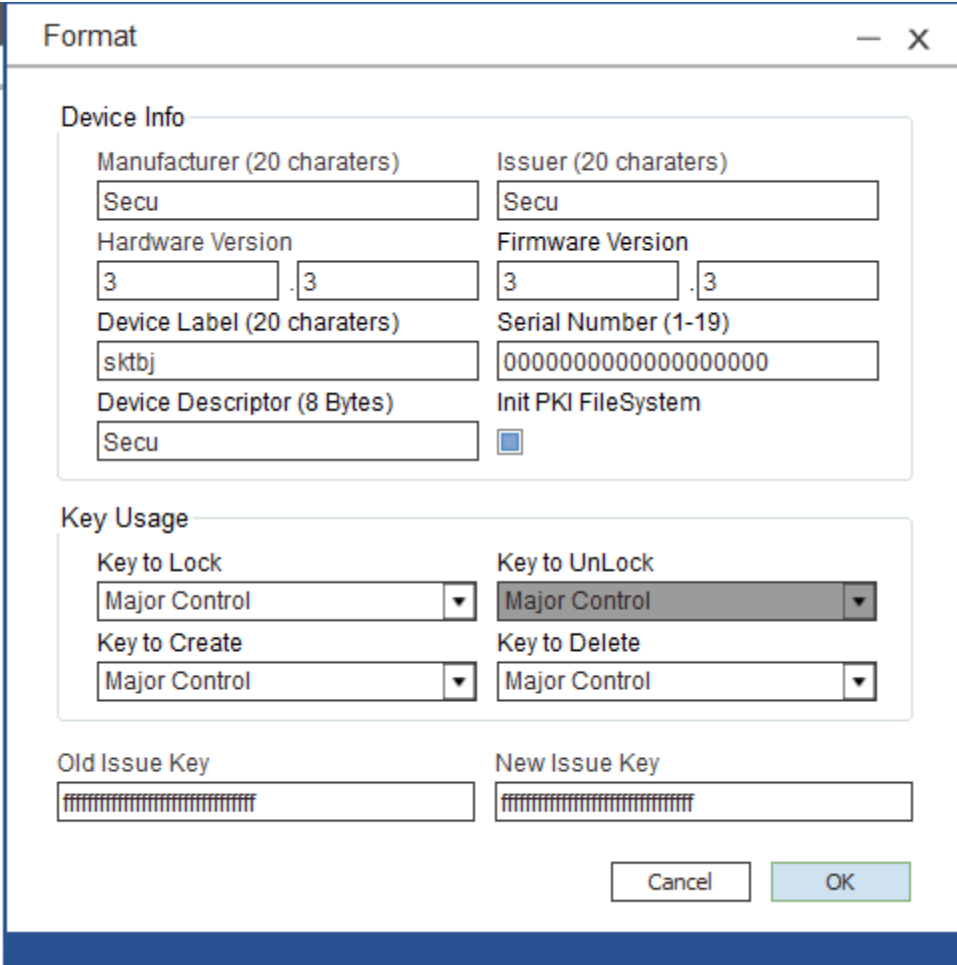


4.3.2 **Initialize Token**

1. On the main page, select the Token from the list.



2. On the left side, click on the initialization icon. In the pop up page, fill the information, configure the key usage and input old issue key and set new issue key.

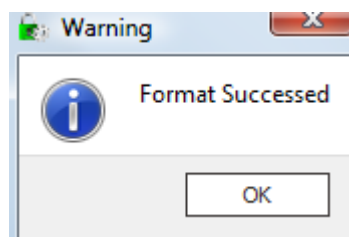


The image shows a 'Format' dialog box with the following fields and options:

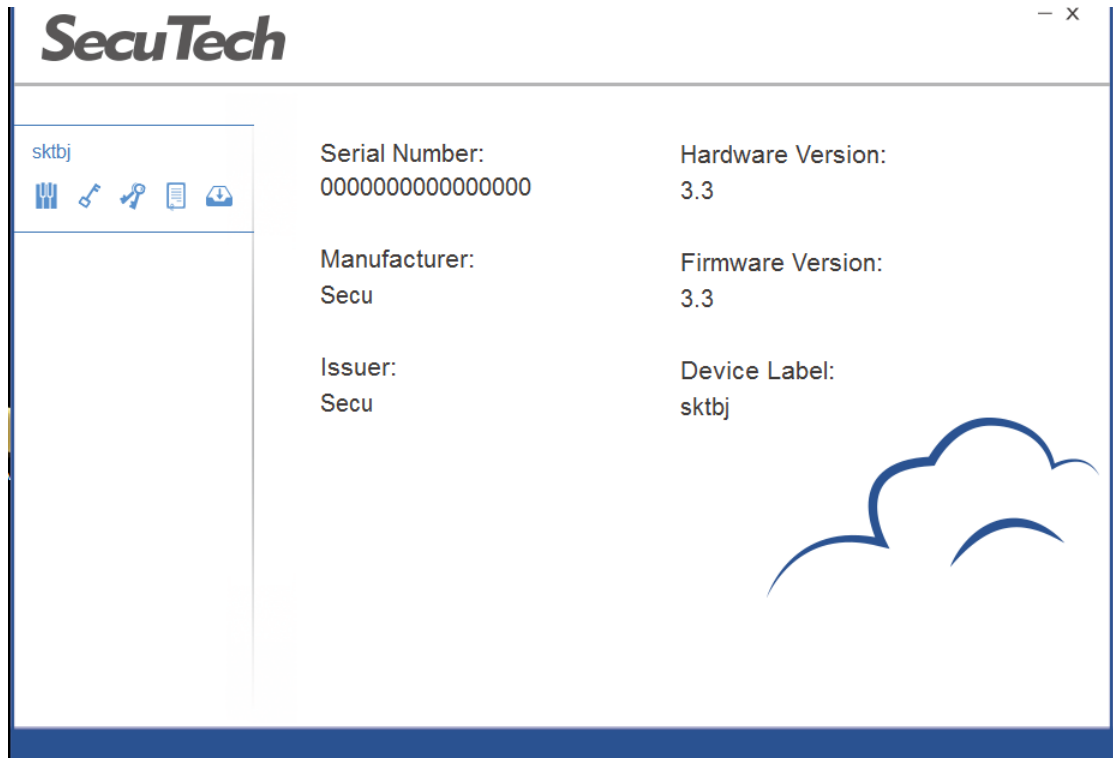
- Device Info:**
  - Manufacturer (20 characters): Secu
  - Issuer (20 characters): Secu
  - Hardware Version: 3 . 3
  - Firmware Version: 3 . 3
  - Device Label (20 characters): sktbj
  - Serial Number (1-19): 00000000000000000000
  - Device Descriptor (8 Bytes): Secu
  - Init PKI FileSystem:
- Key Usage:**
  - Key to Lock: Major Control
  - Key to UnLock: Major Control
  - Key to Create: Major Control
  - Key to Delete: Major Control
- Old Issue Key:** [Masked]
- New Issue Key:** [Masked]

Buttons: Cancel, OK

3. Click on OK to start initialization.

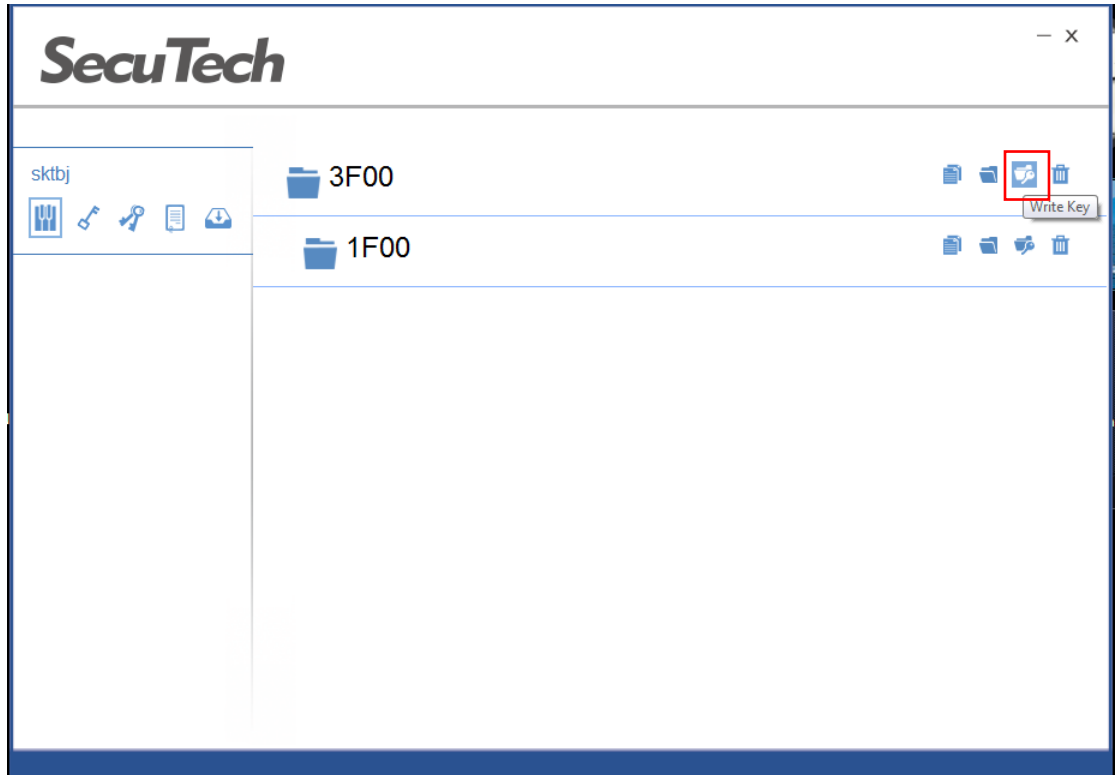


After the token is initialized successfully, a message page will pop up. Click on OK to return the main page.

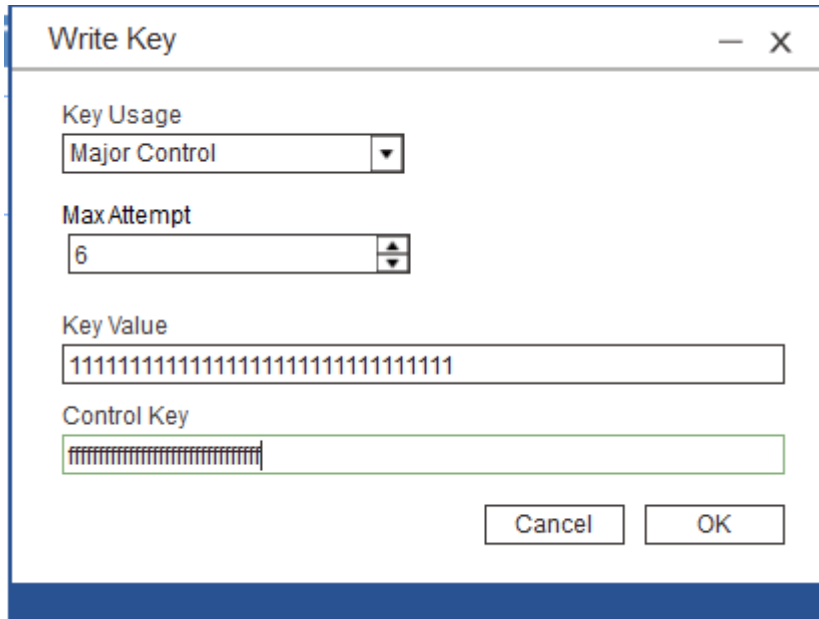


#### 4.3.3 **Change Key**

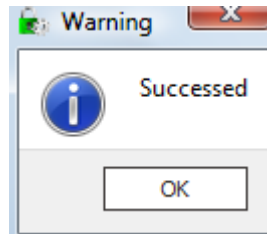
1. Write a key for the folder by clicking on the write key icon.



2. In the pop up page, select the key usage, input the key value, the maximum attempts and input the master key of the folder.



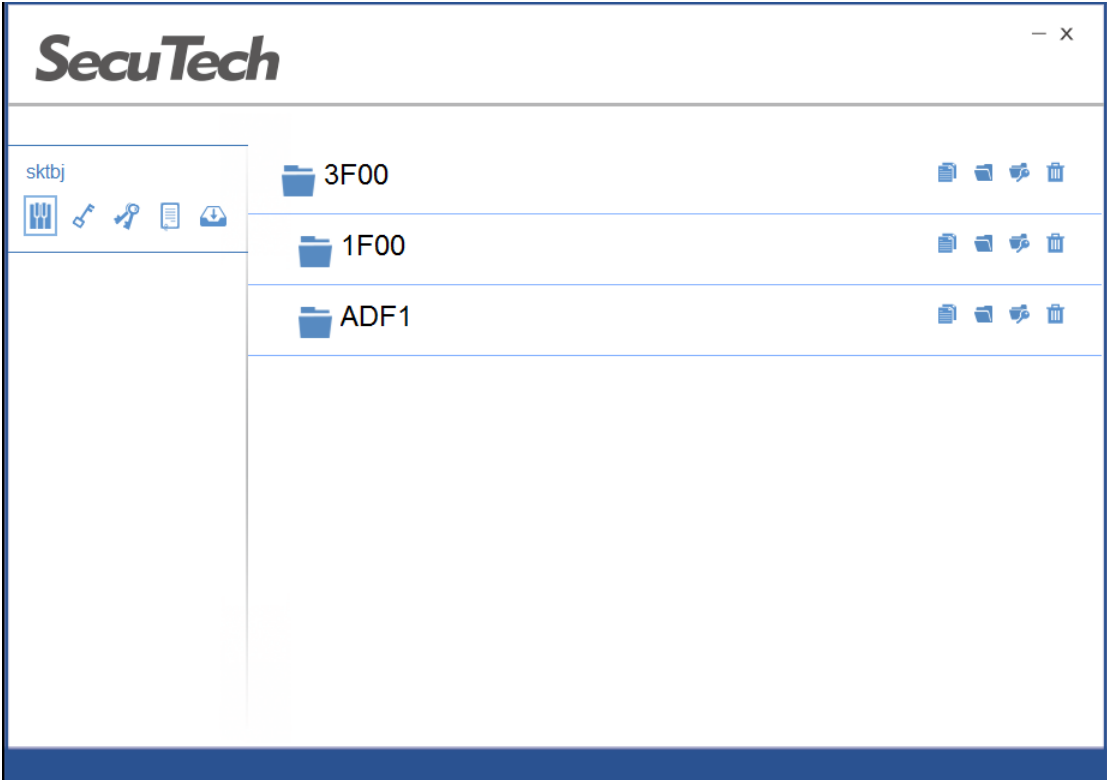
3. Click on OK



#### 4.3.4 **Create folder (max 3 level)**

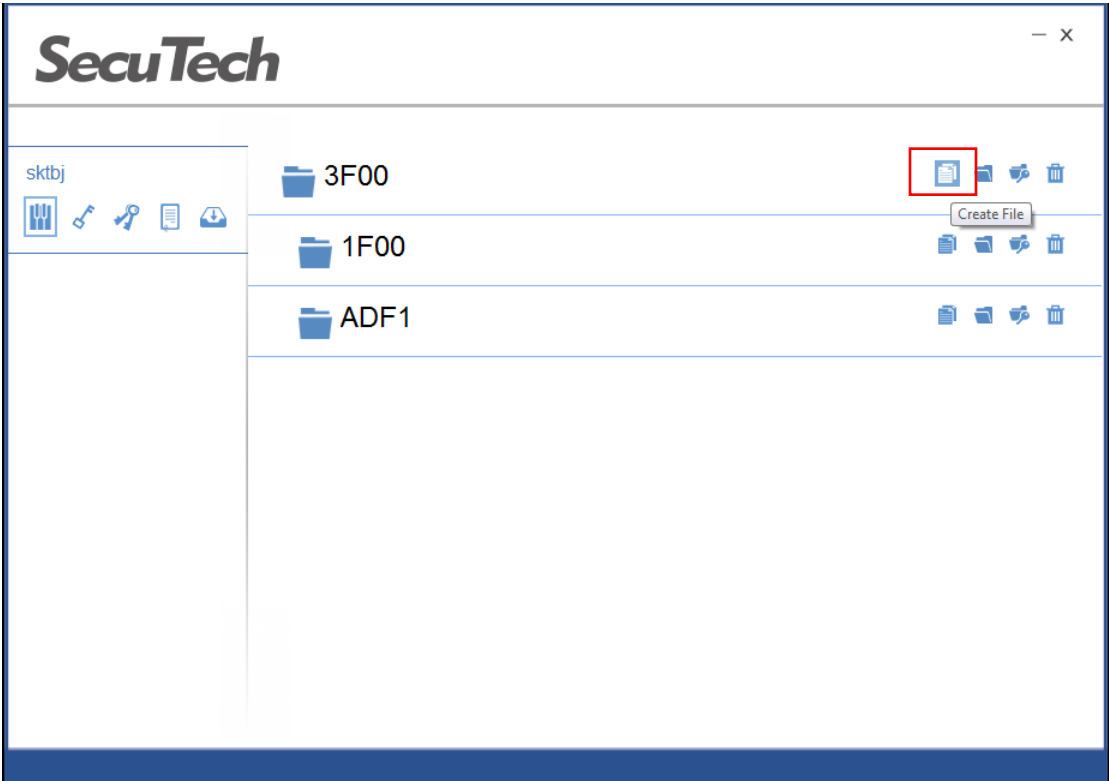
1. Click on the create folder icon





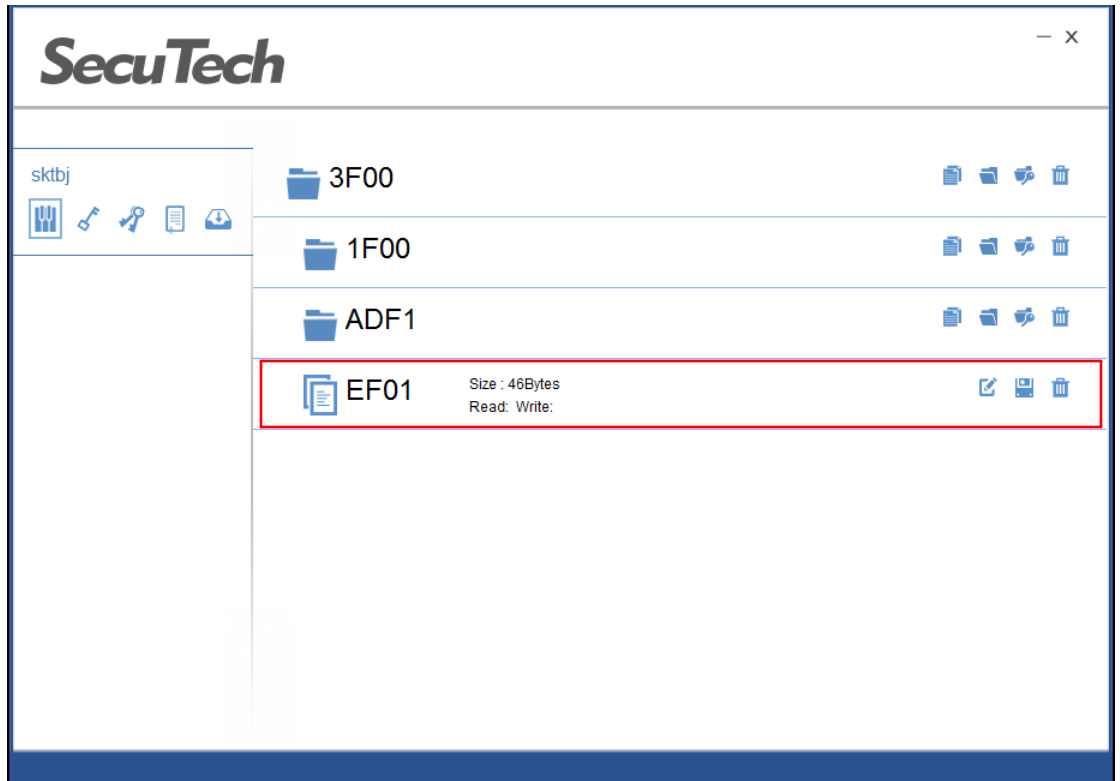
4.3.5 **Create file**

Click on the create file icon under the selected folder.





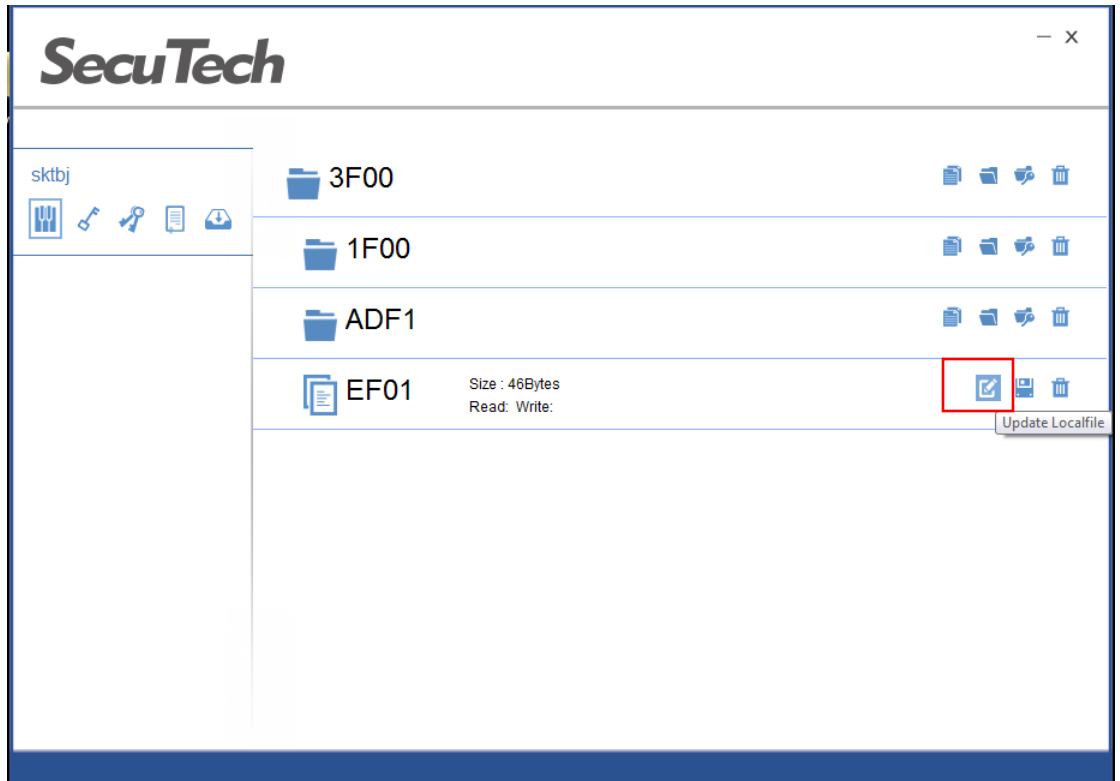




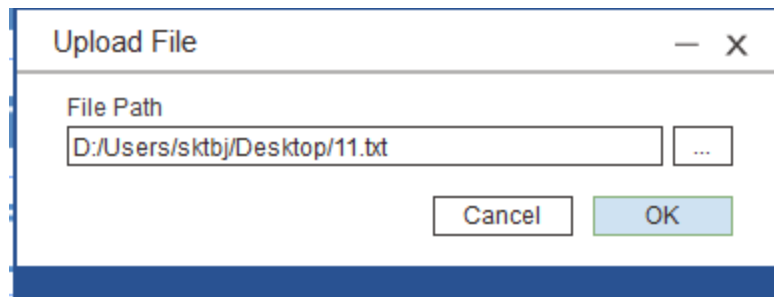
#### 4.3.6 **Read/write file**

Write file

1. Select the file and click on the update local file icon.



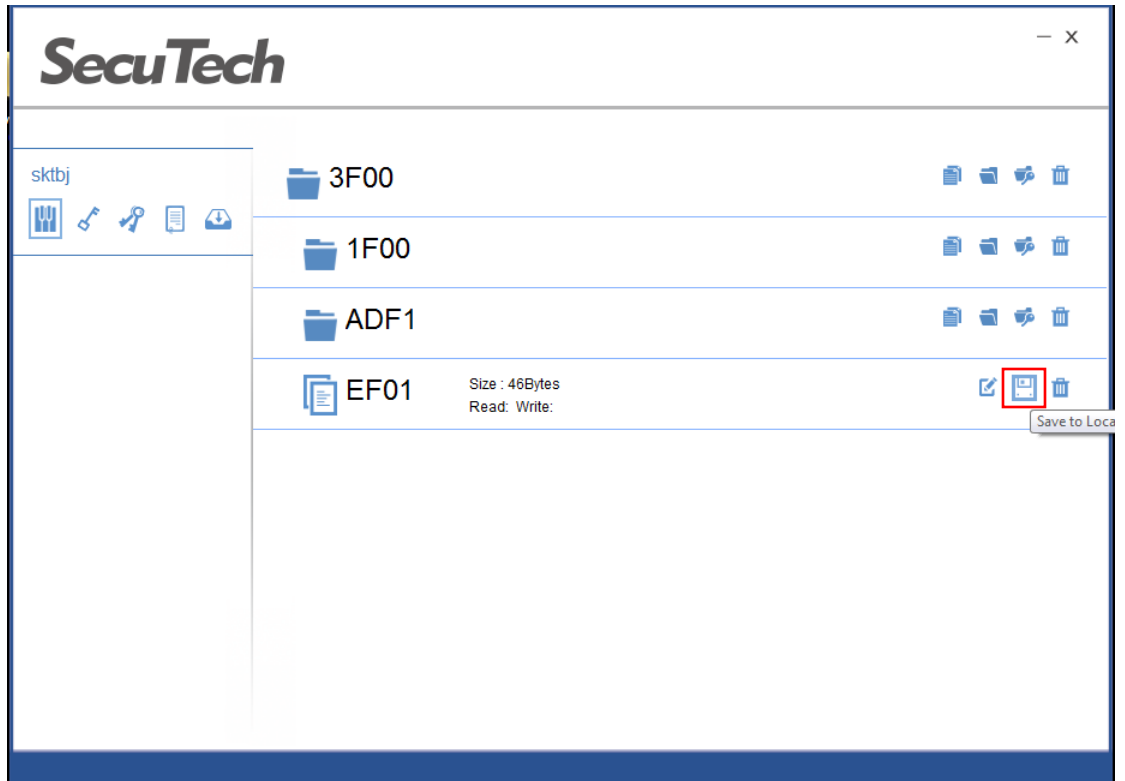
In the pop up page, select the file from your PC



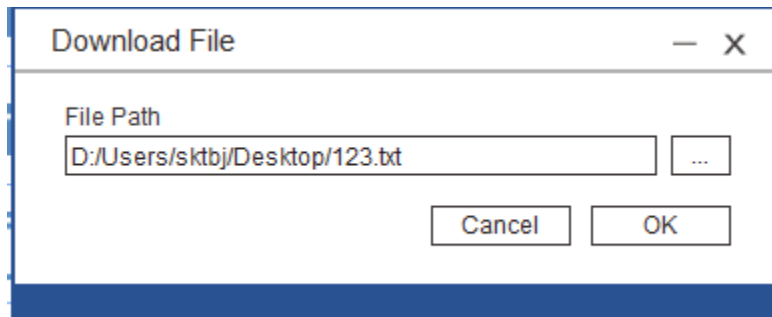
The token will authenticate the write right of the selected file according to the file access right configured when the file is created.

Read File

1. Select the file and click on the Save to local icon.



2. In the pop up page, input the directory and file name that the selected file to be saved.



3. Click on OK to save the file in token to the local PC.

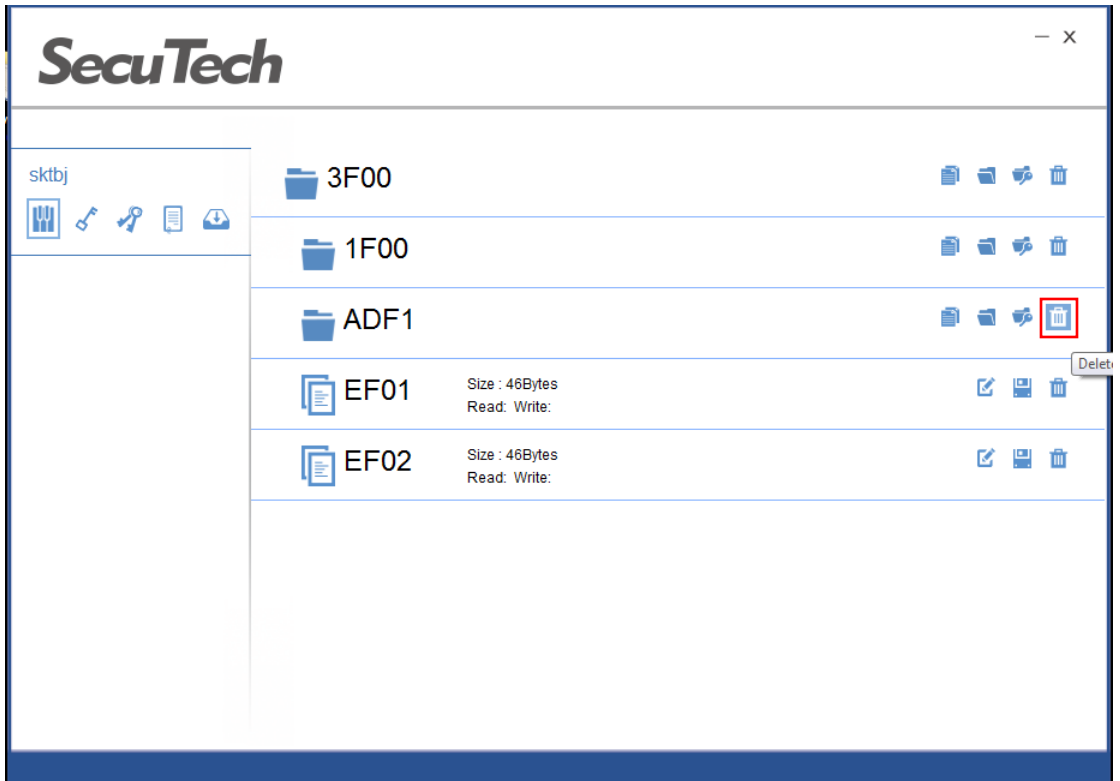
The token will authenticate the read right according to the access right configured when the file is created.

The file will be found in your local PC after it's saved successfully.

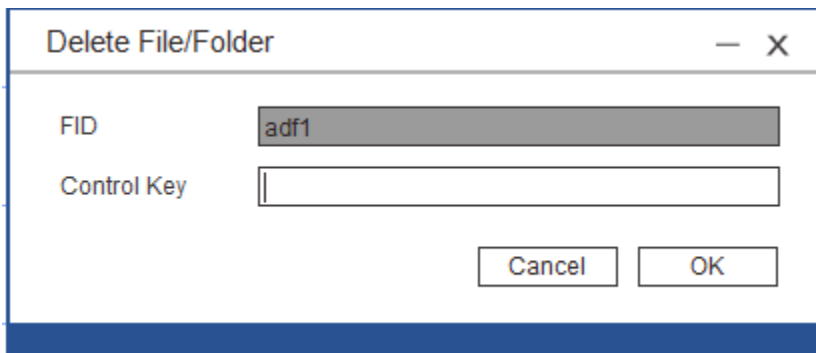
#### 4.3.7 **Delete file/ folder**

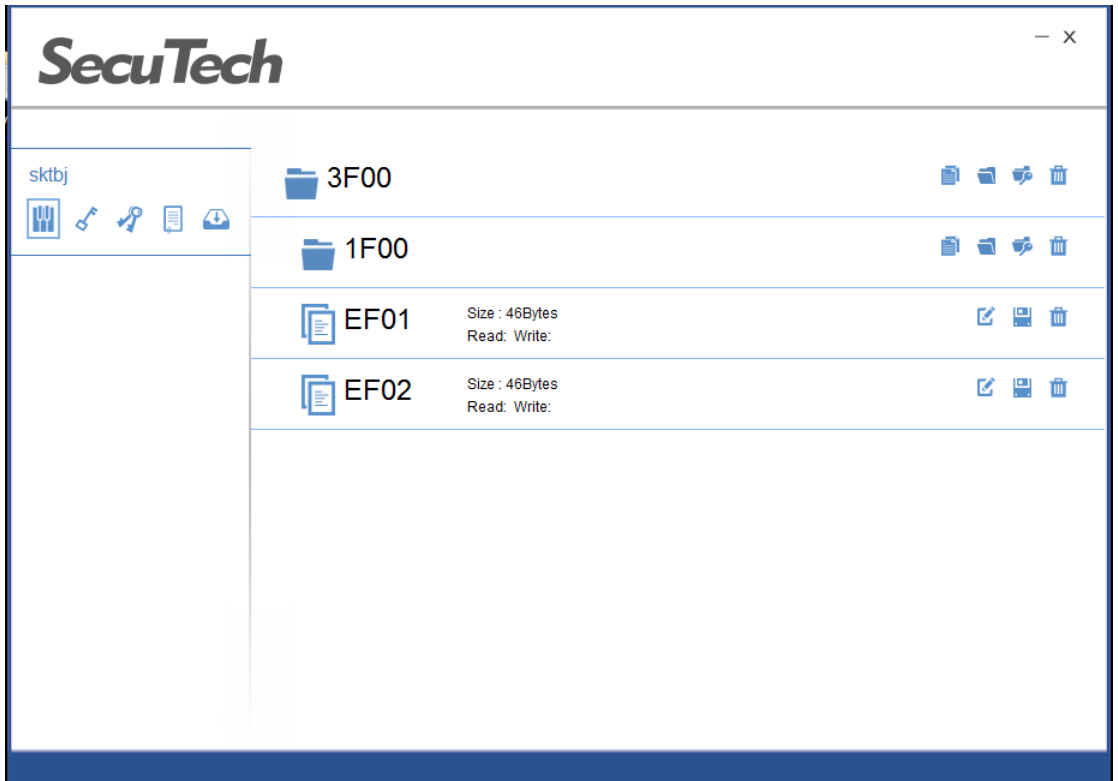
Delete folder

1. Select the folder and click on the delete icon.



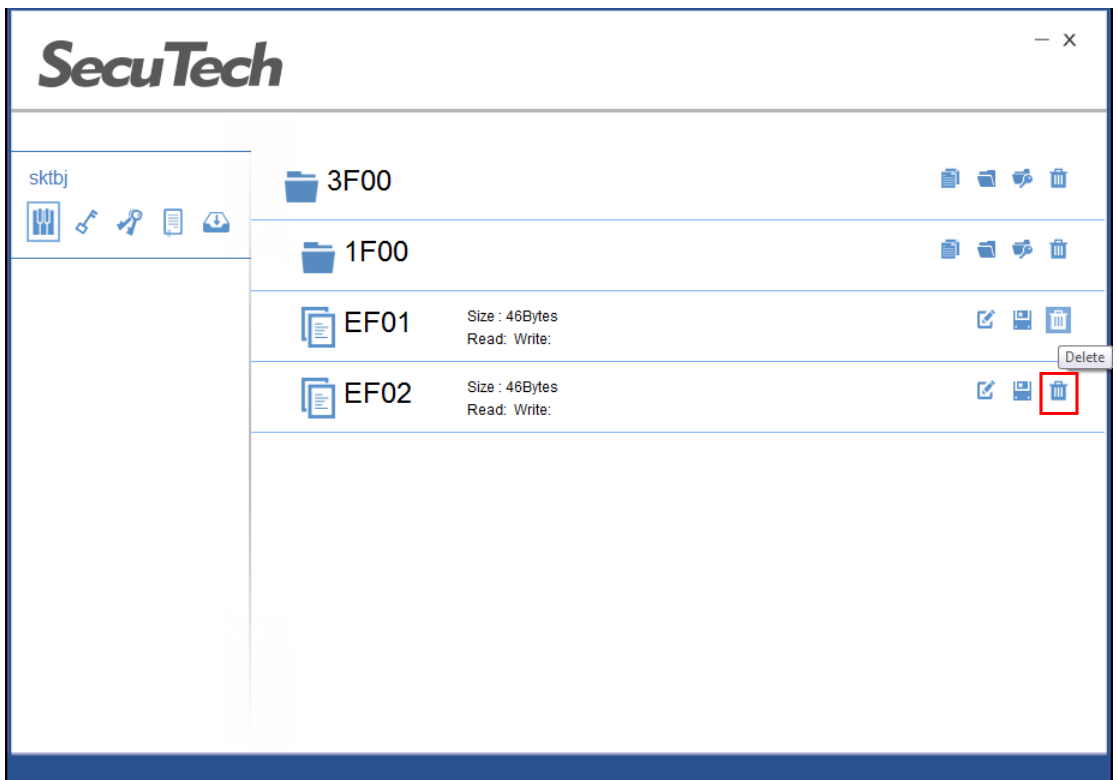
2. Input the key of the upper level of the selected file and click on OK



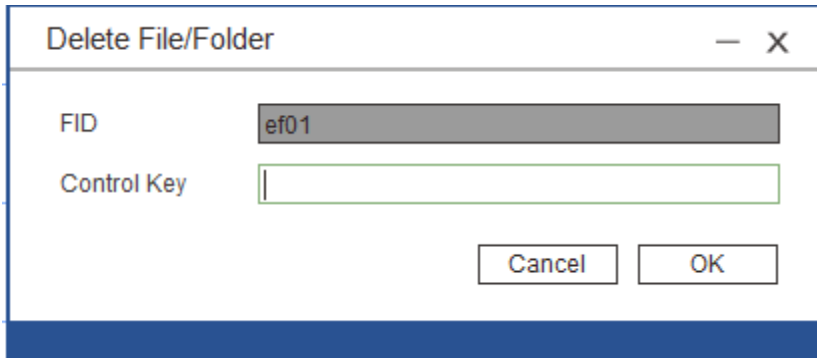


Delete file

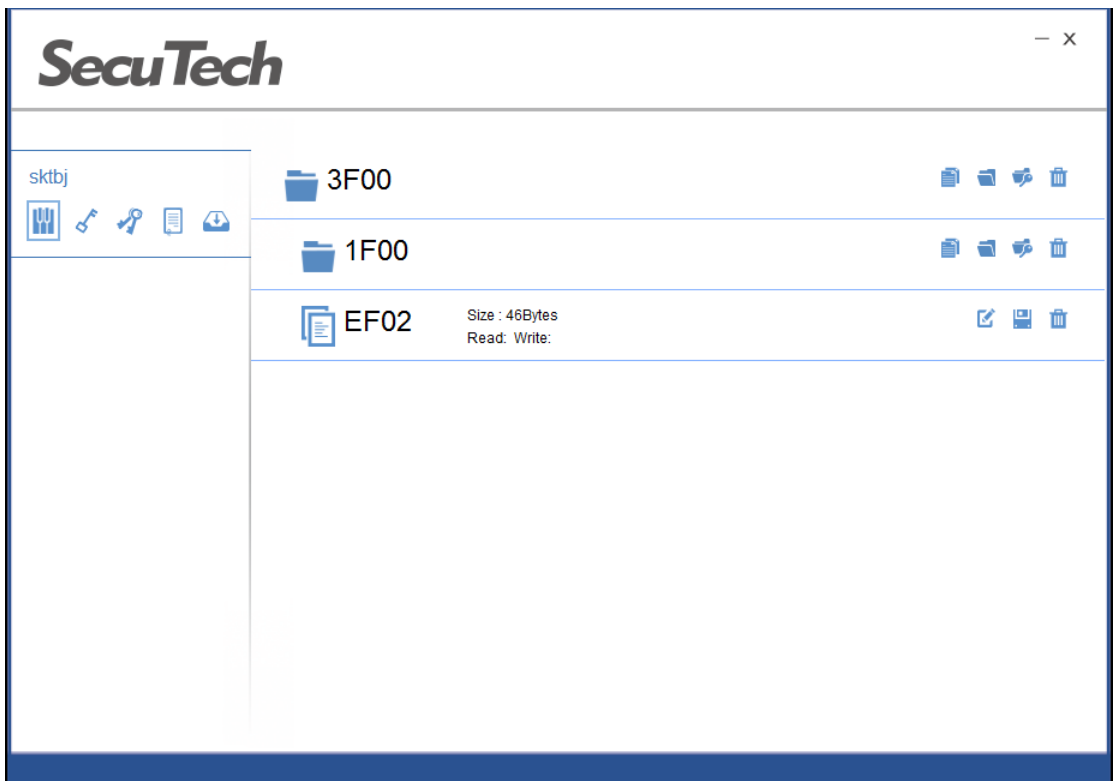
1. Select the file to be deleted, and click on the delete icon



2. In the pop up page input the key of the upper folder and click on OK.

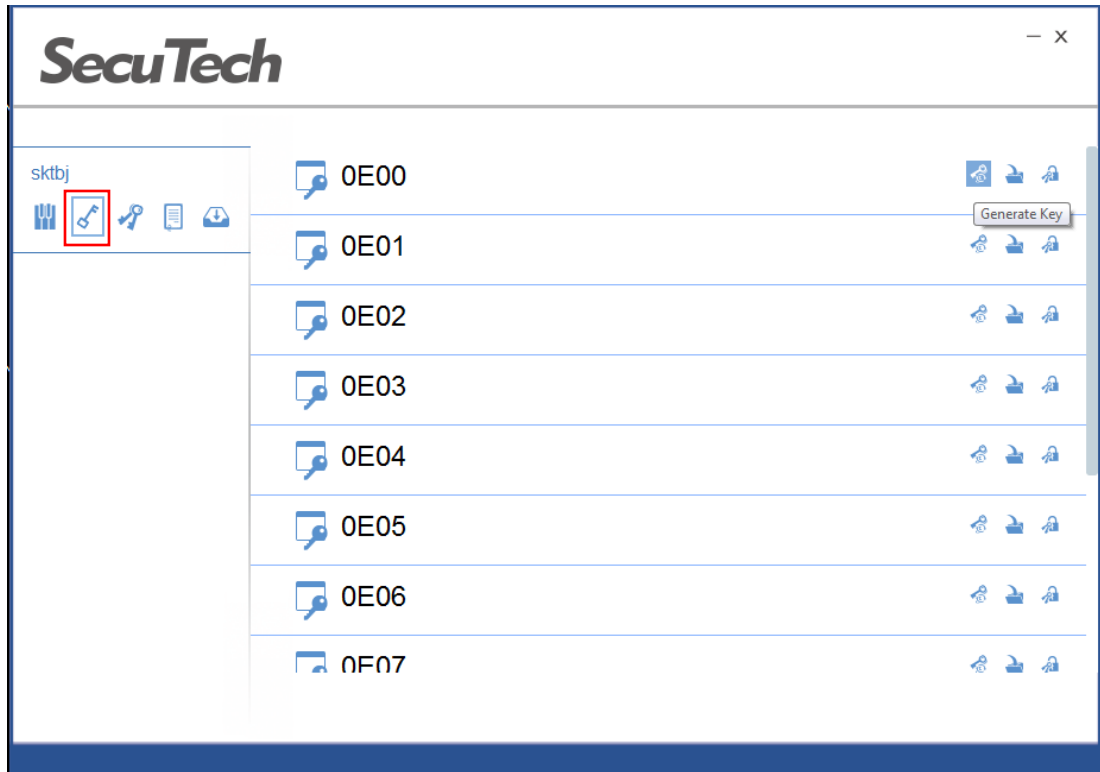


The selected file will disappear after it's deleted successfully.



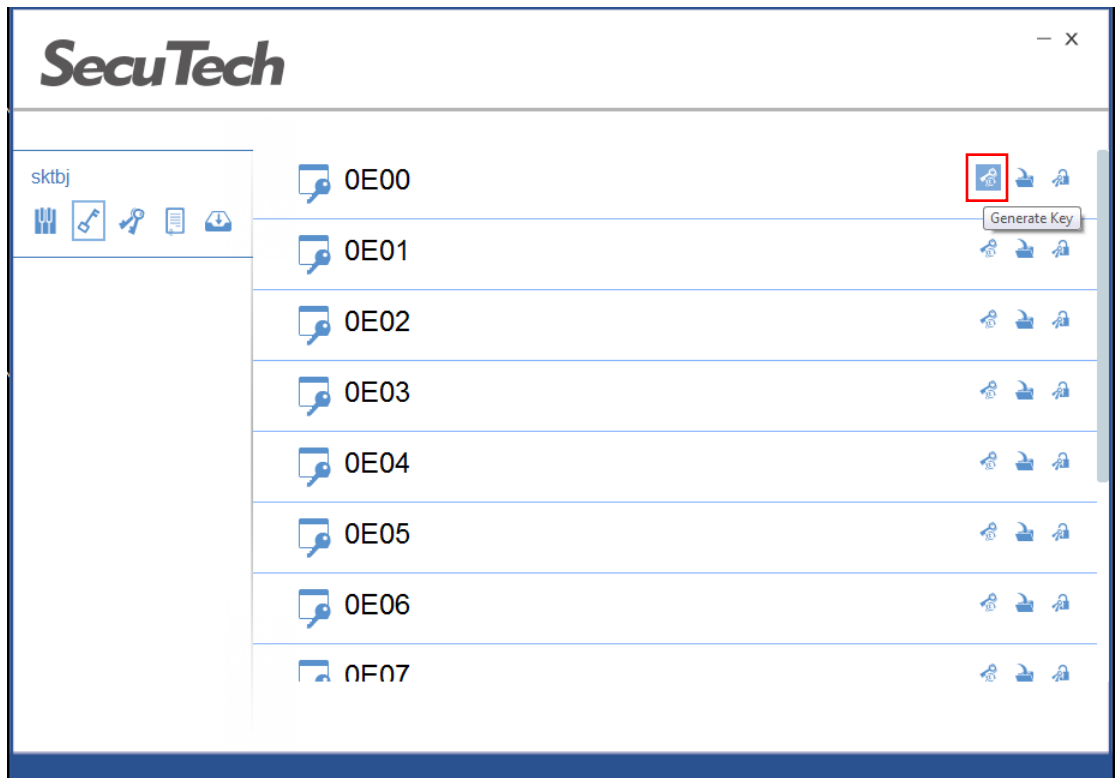
#### 4.3.8 **Symmetric Key**

Click on the symmetric keys icon.



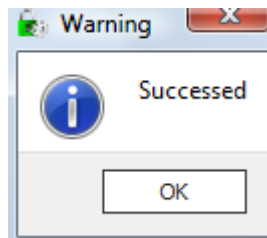
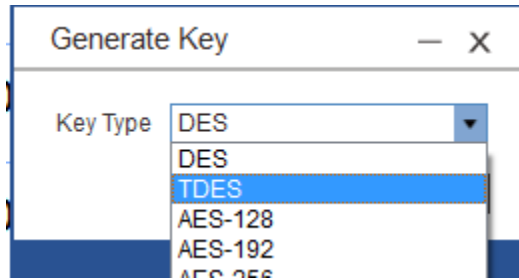
Generate Key

1. Select a key file from the list and click on the generate key icon



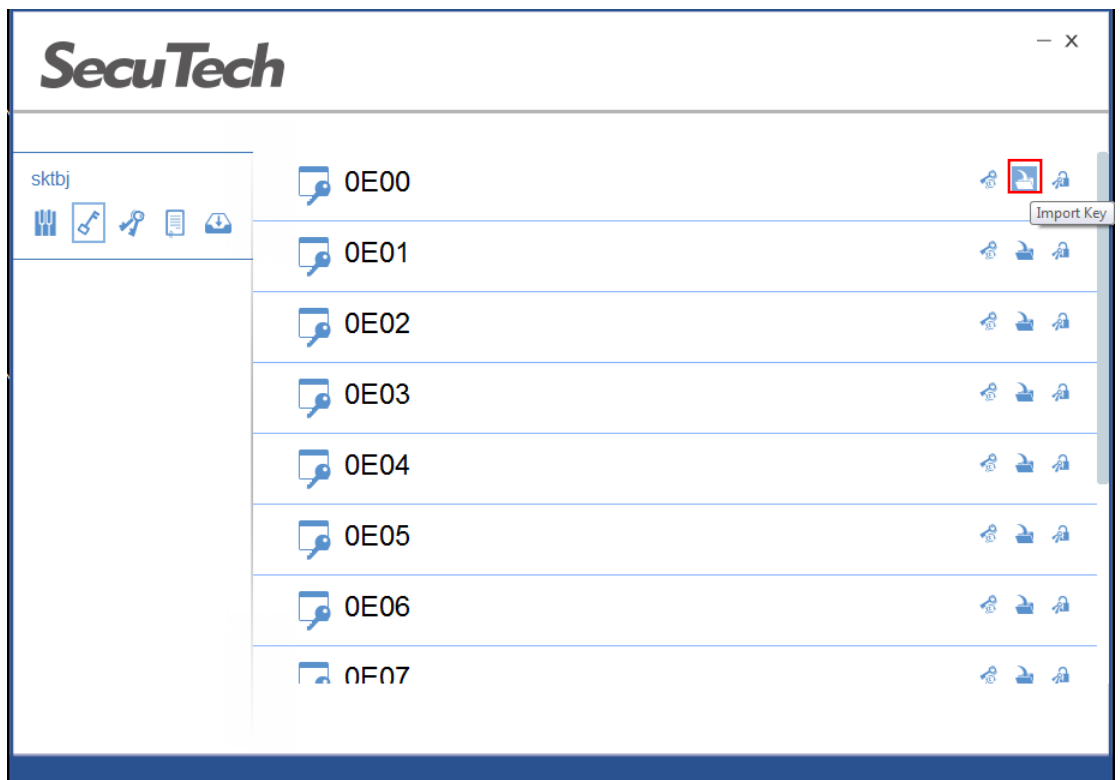
2. In the pop up page, select key type and click on OK.



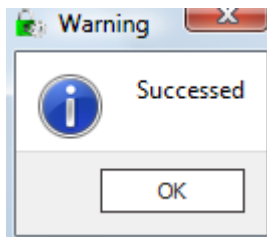
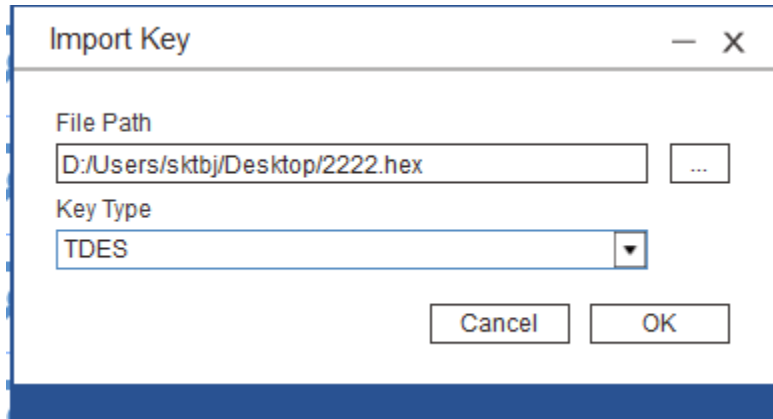


- Import Key

1. Select a key file and click on the import key icon.

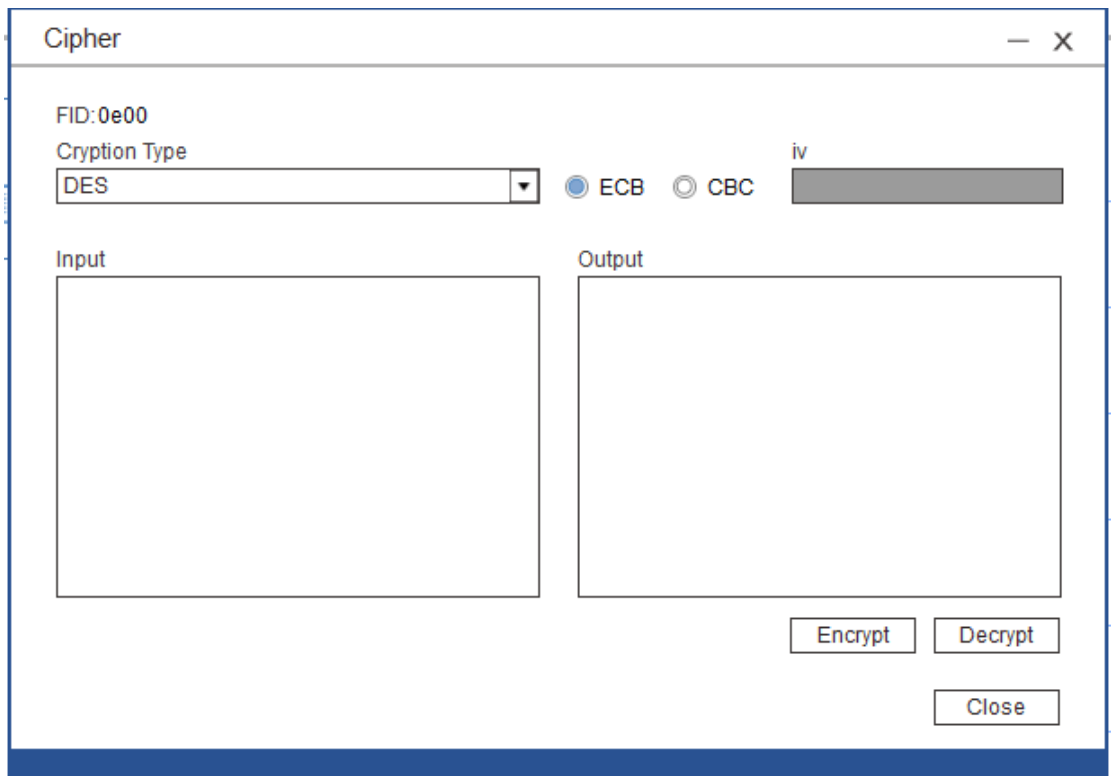


2. In the pop up page, find your key file and select key type. Click on OK.

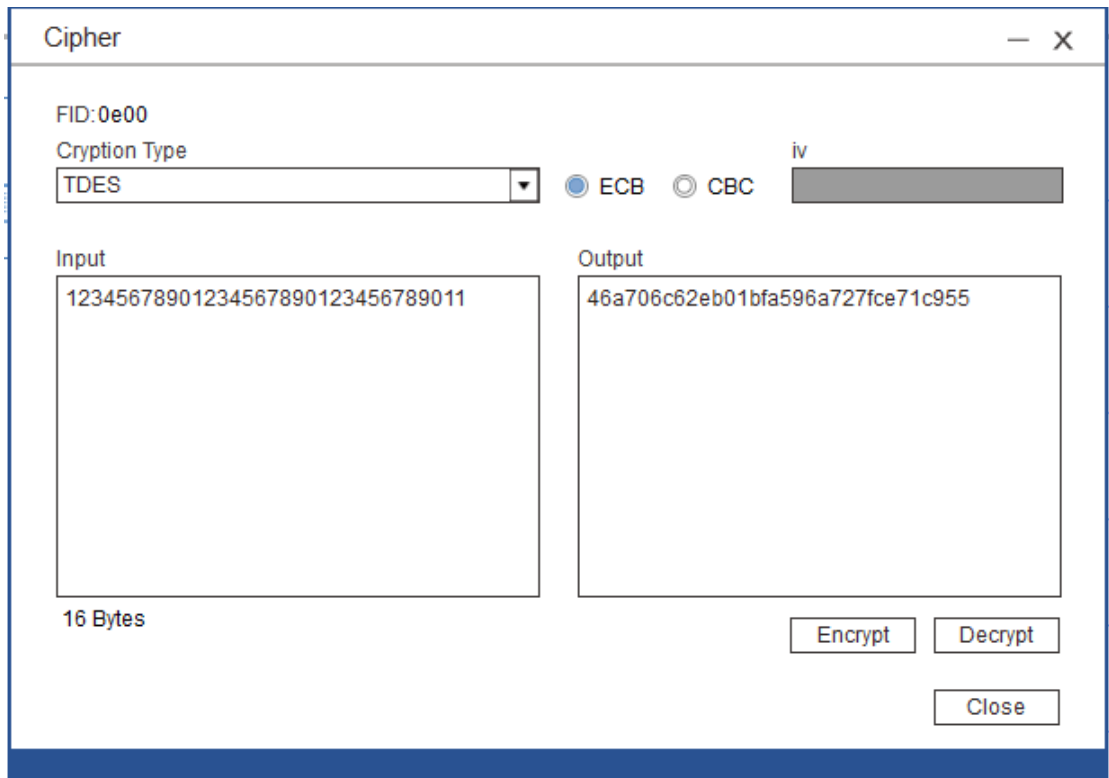


- Encrypt/decrypt

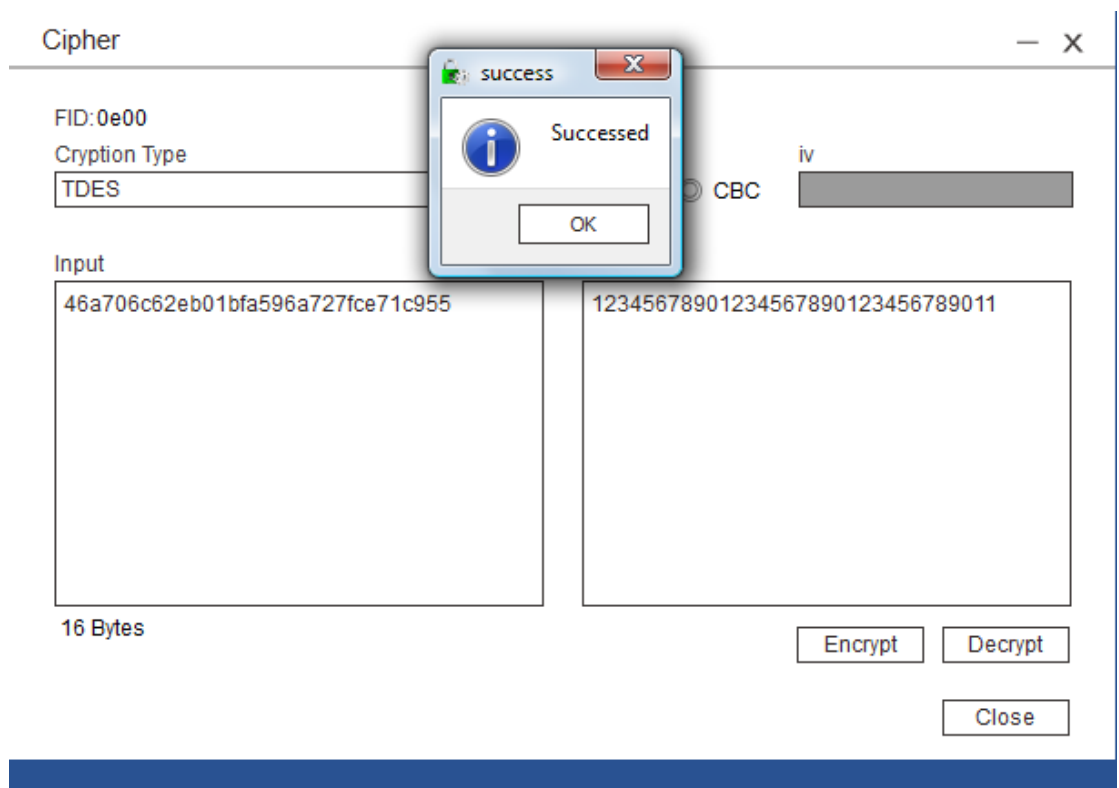
Select the key file to be used to encrypt/decrypt. (ensure a key has been stored in the selected key file and key type)



Select the algorithm and input the data in HEX to be encrypted/decrypted. Click on the encrypt/decrypt button.

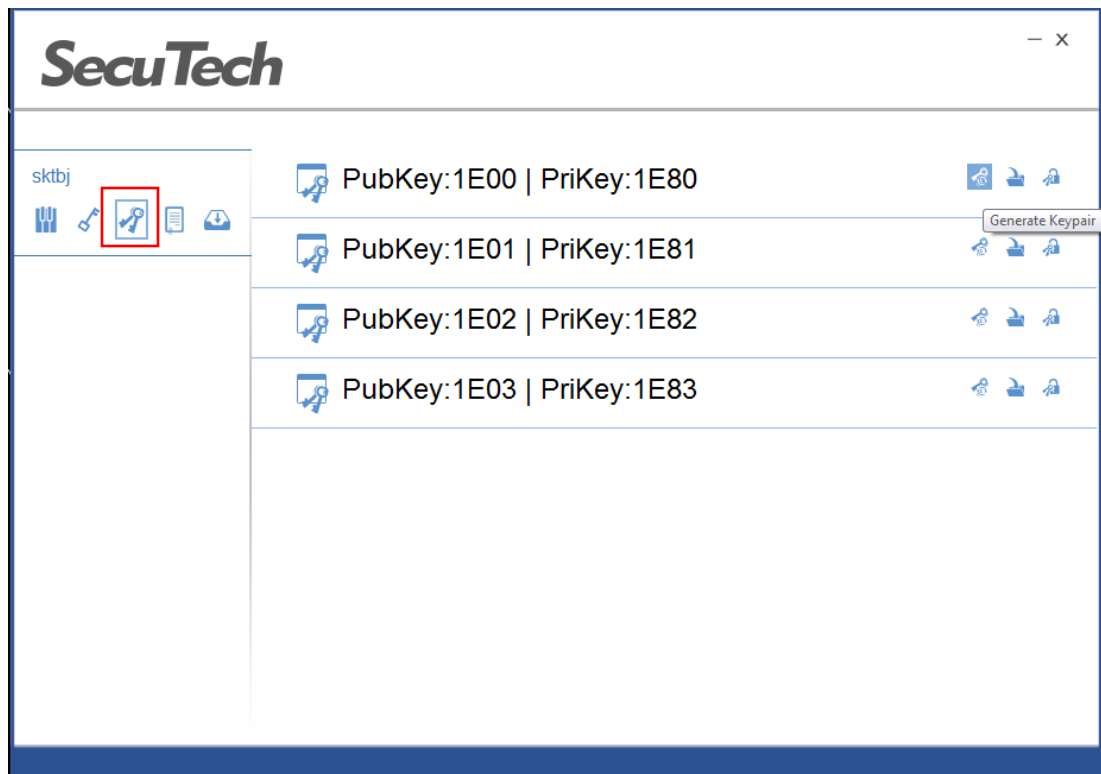


Result will display in the output box.

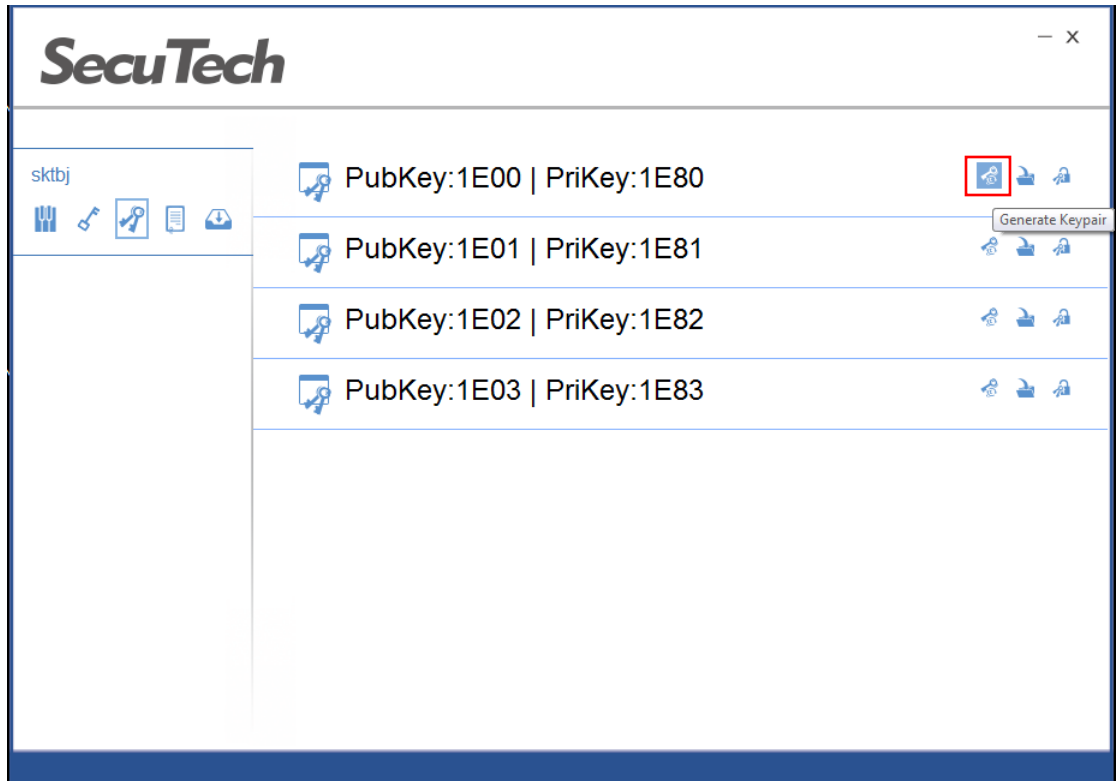


#### 4.3.9 Asymmetric Key

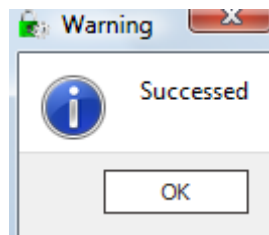
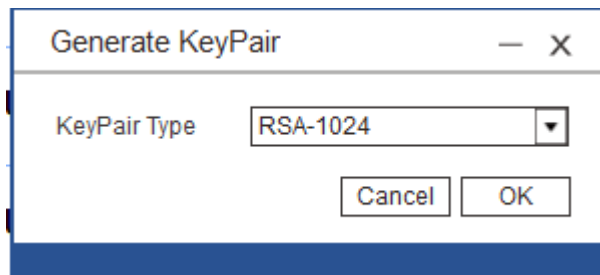
Select the symmetric key pair icon



- Generate key
  1. Select a key file from the list and click on the generate key icon.

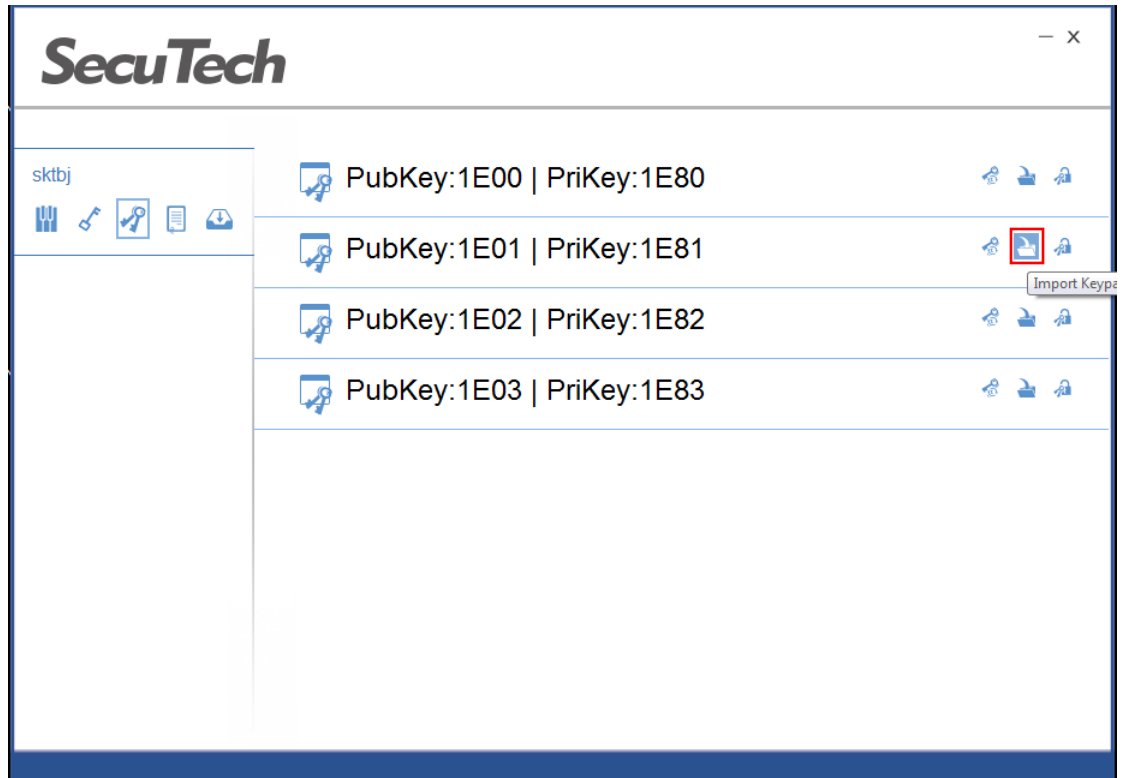


2. In the pop up page select key pair type and click on OK

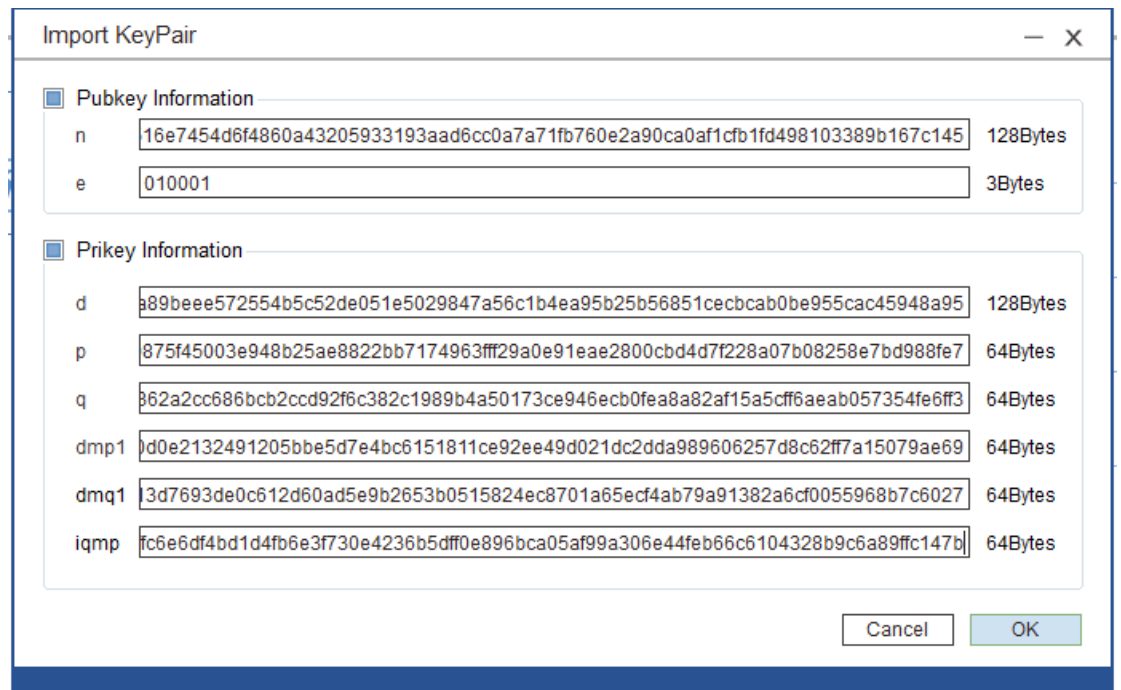


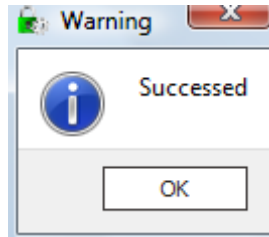
- Import Key Pair

1. Select key file and click on the import icon.

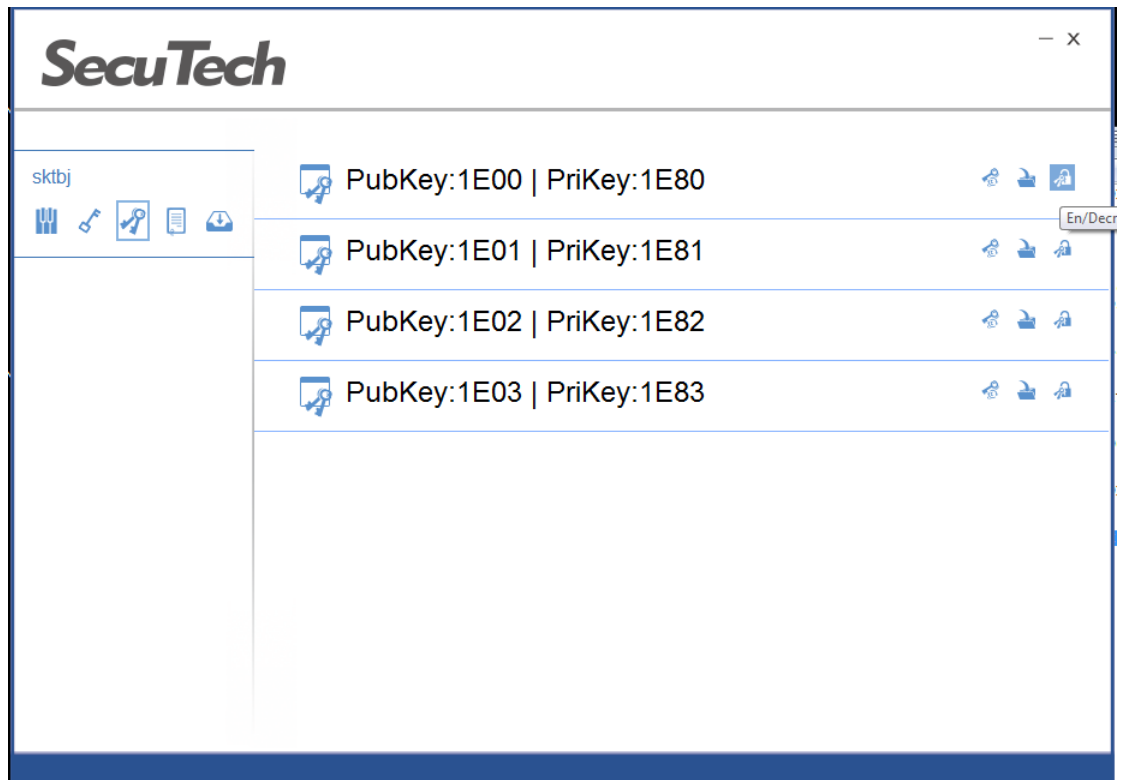


2. In the pop up page, input the correct key, and click on OK



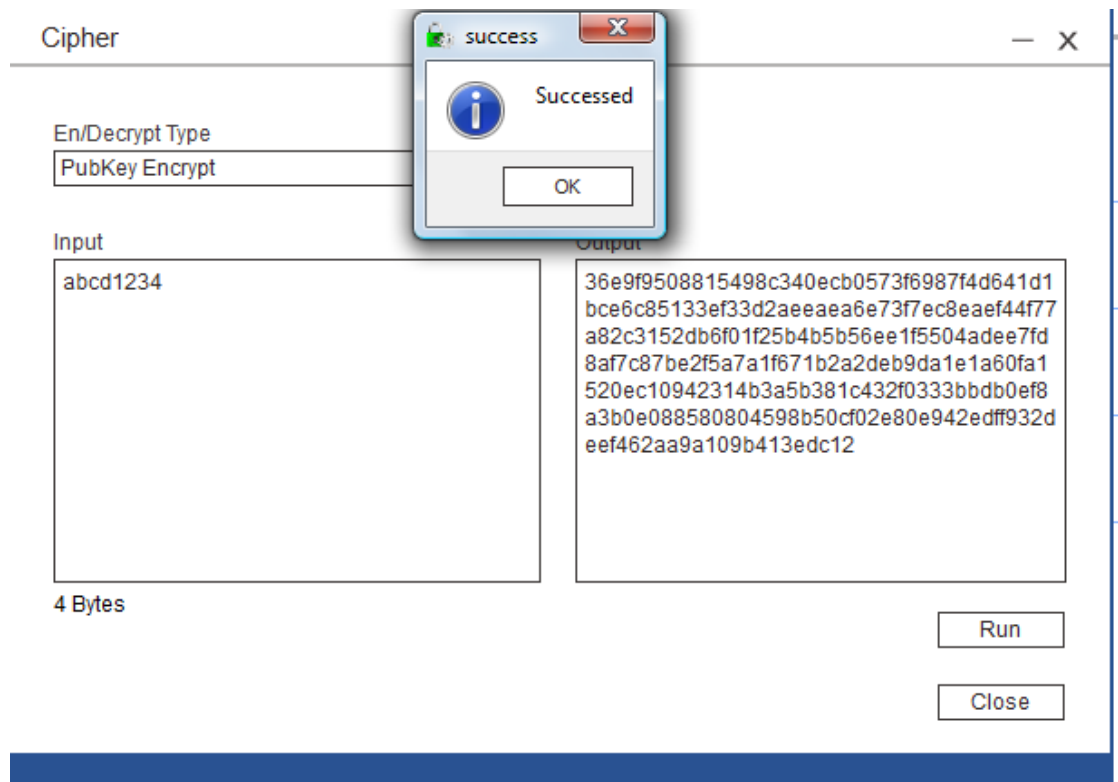


- Encrypt/decrypt
  1. Select the key file

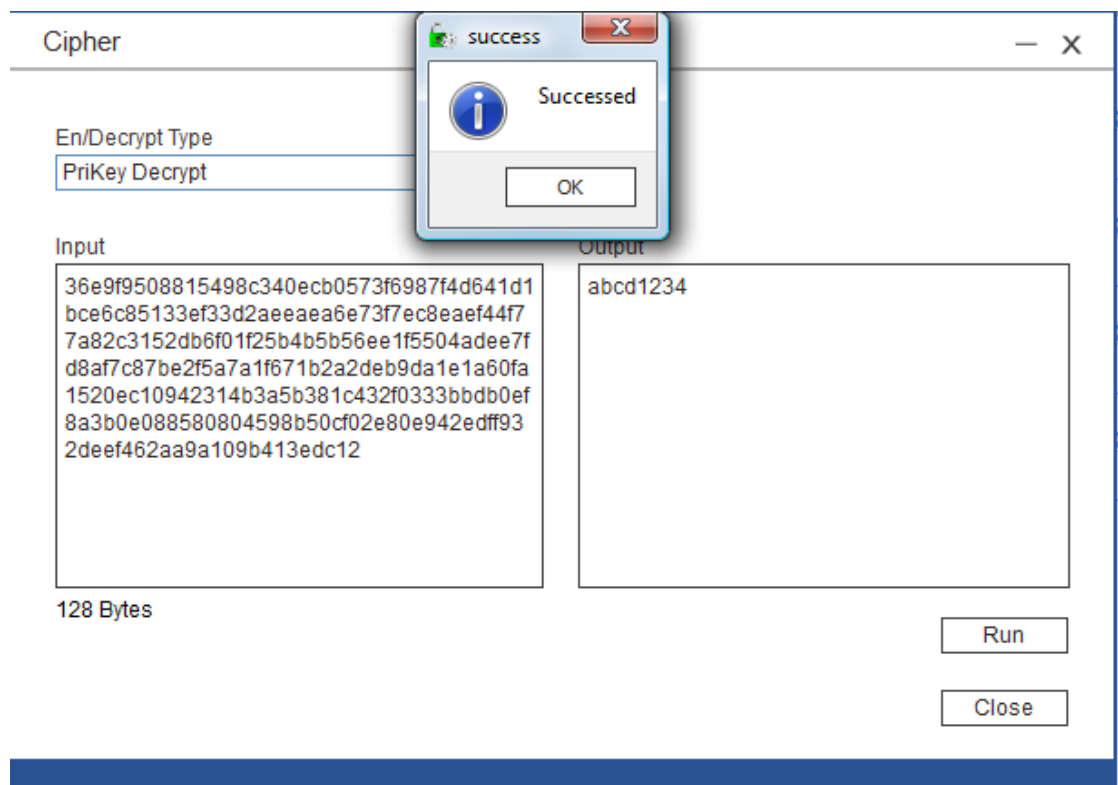


2. Select encrypt

Input data in HEX to be encrypted and click on run.



In the same procedure, select decrypt and click on run to decrypt.

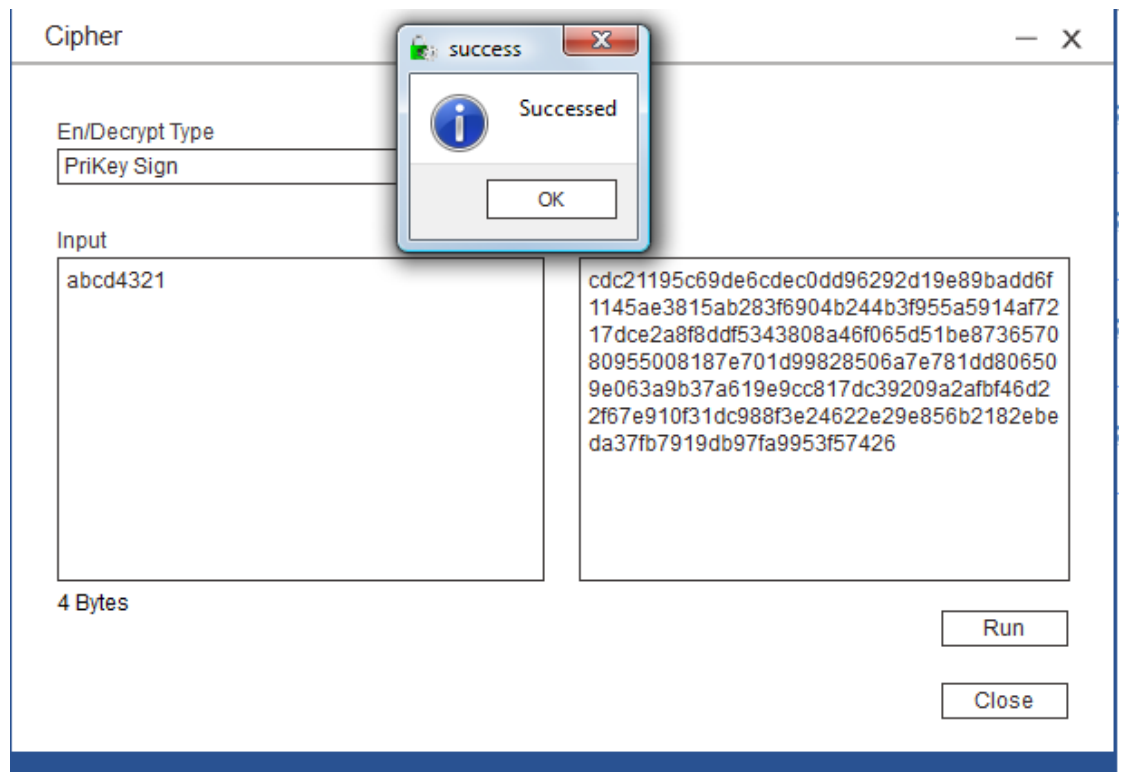


Sign



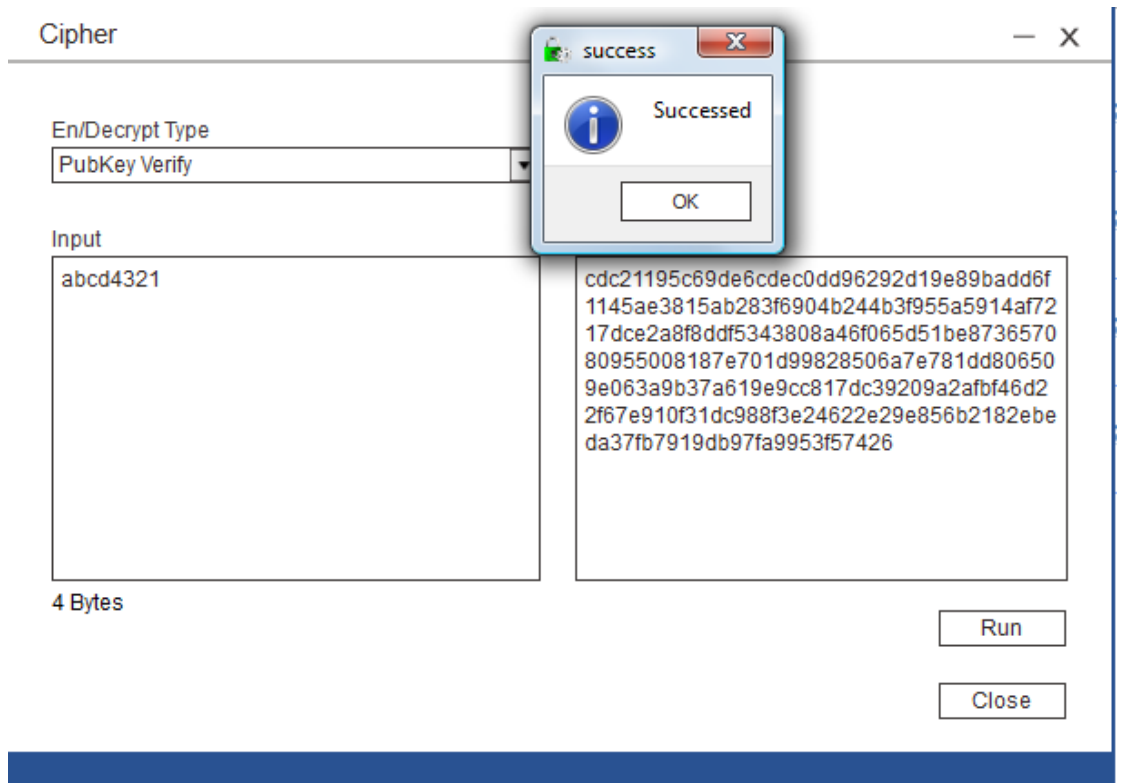
Select sign

Input data in HEX to be signed and click on run.



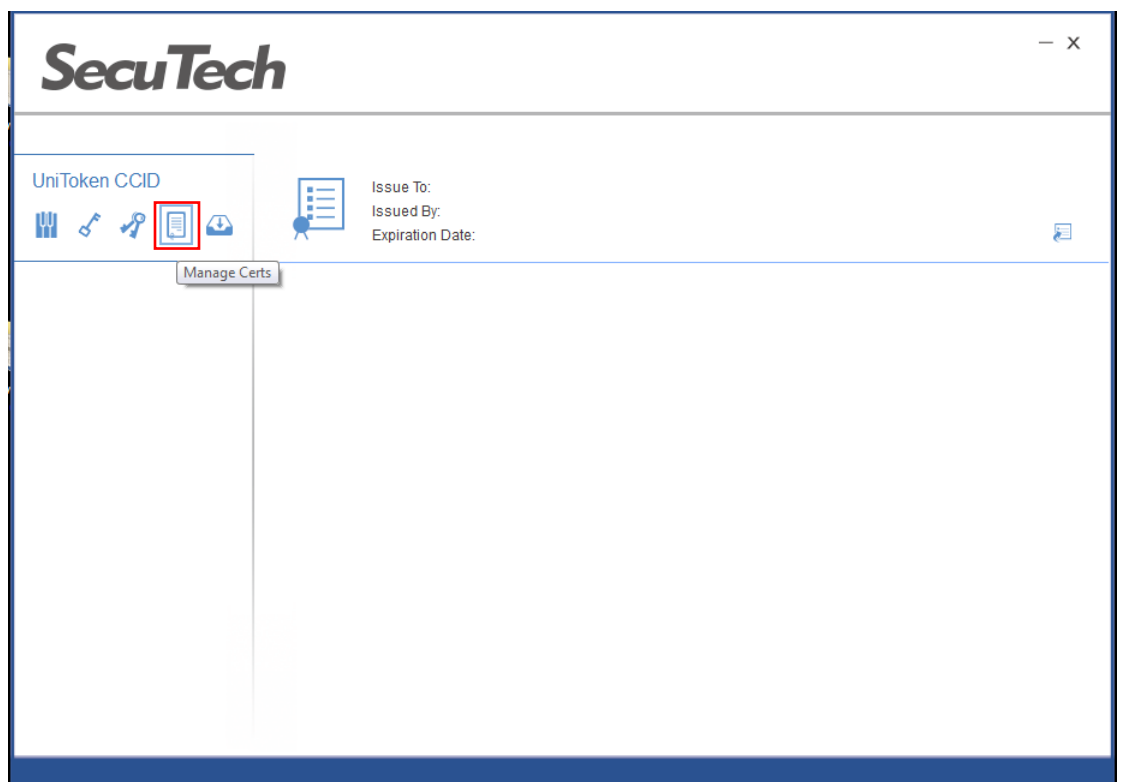
Verify signature

Input the signature, select verify signature and click on run.



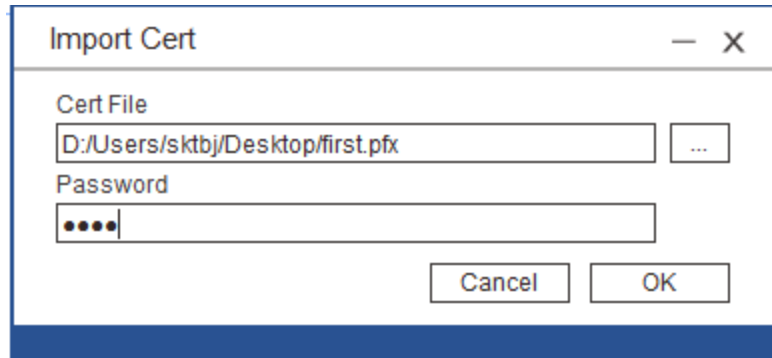
#### 4.3.10 **Change certifiante**

Click on manage certs icon.

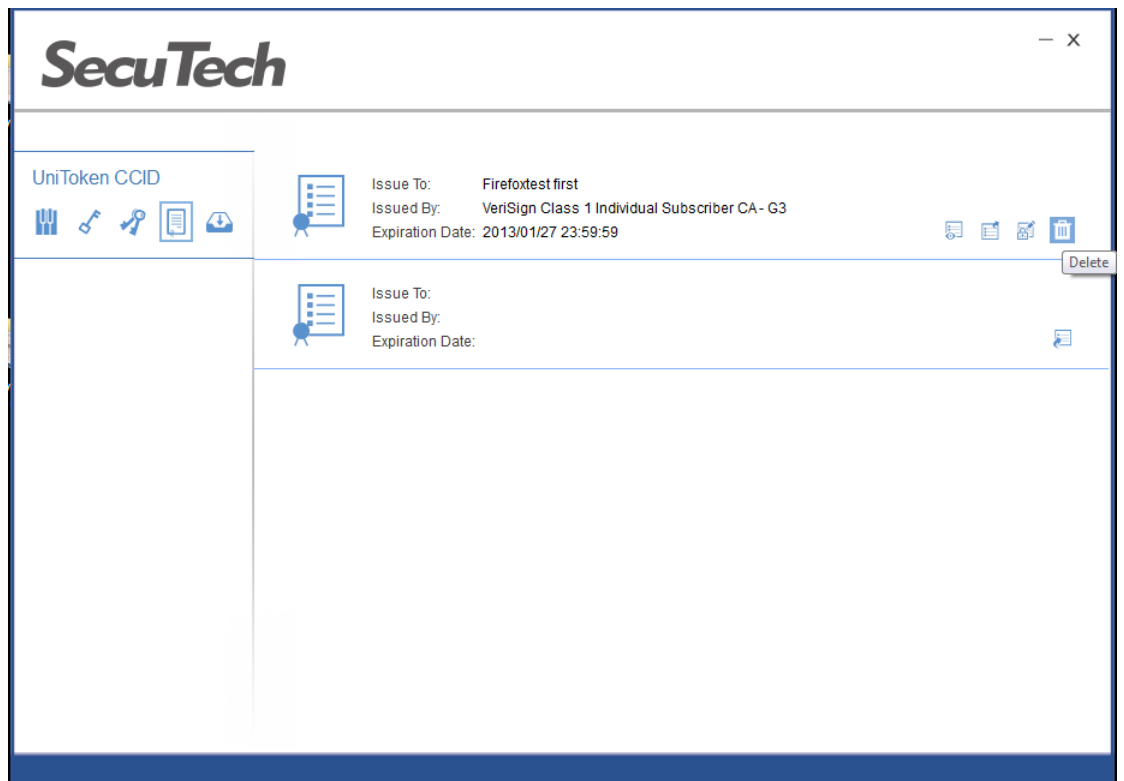


- Import certificate

Click on import certs and select the certificate to be imported and input the password to the certificate.

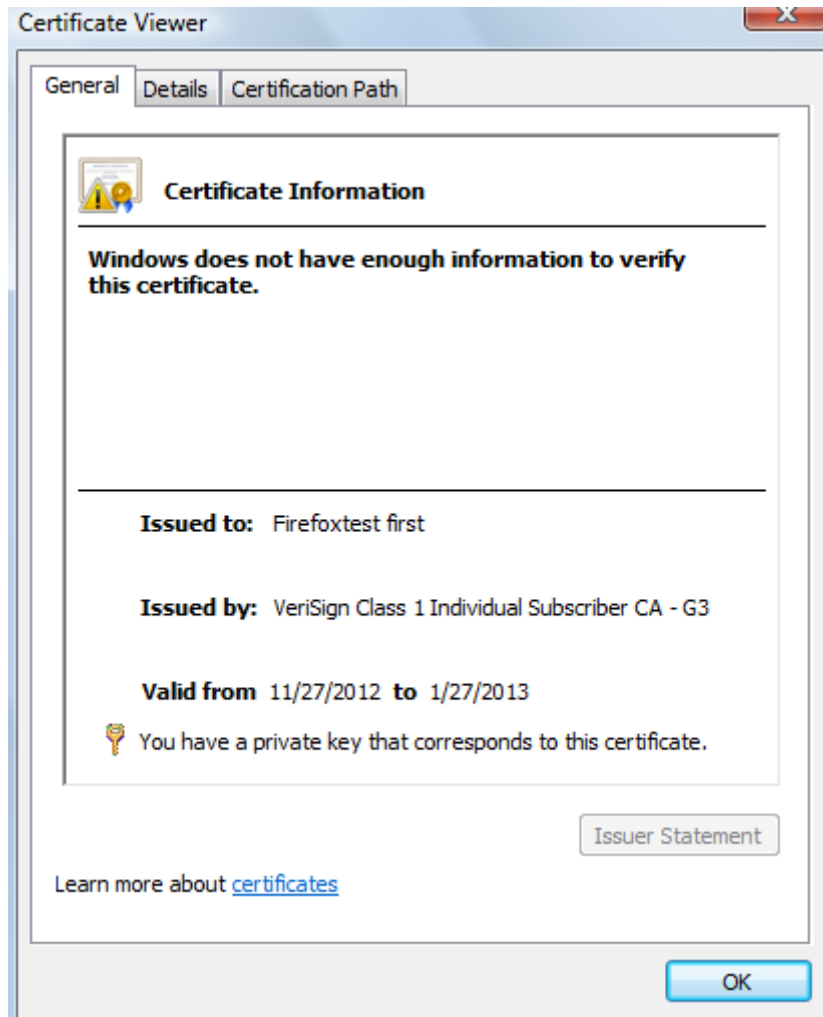


Click on OK to import the certificate.



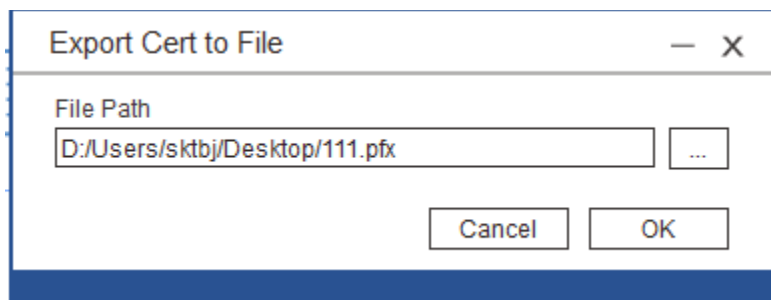
View certificate

Click on the view certs and the certificate information will display in the pop up page.



Export certificate

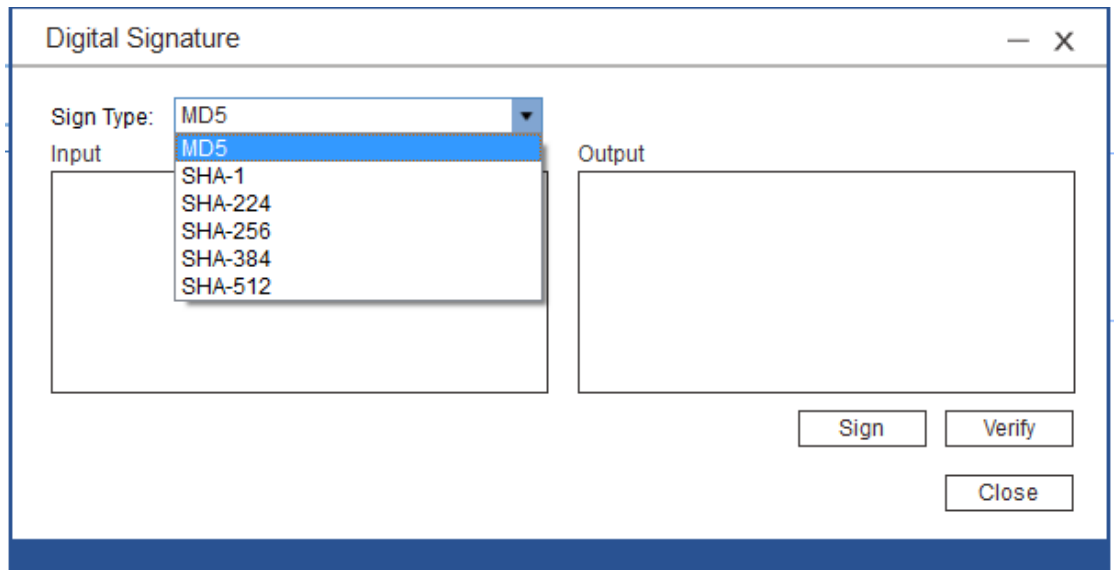
Click on the export certs and specified the directory that the certificate to be saved.



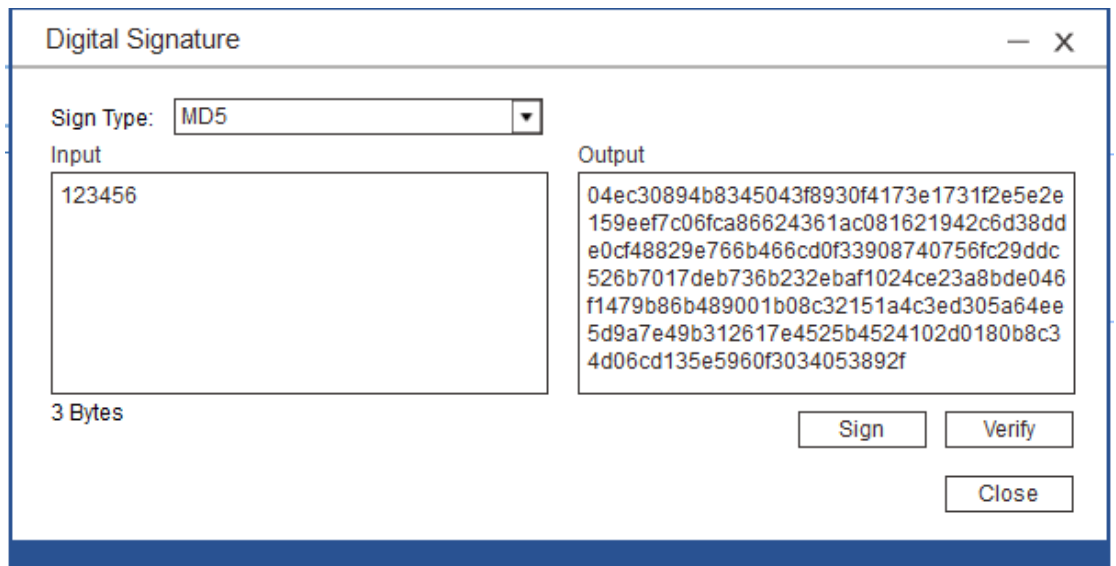
Click on OK and the certificate will be saved to the directory.

- Sign by a certificate

Click on the sign icon and select hash algorithm.



Input data in HEX and click on sign.

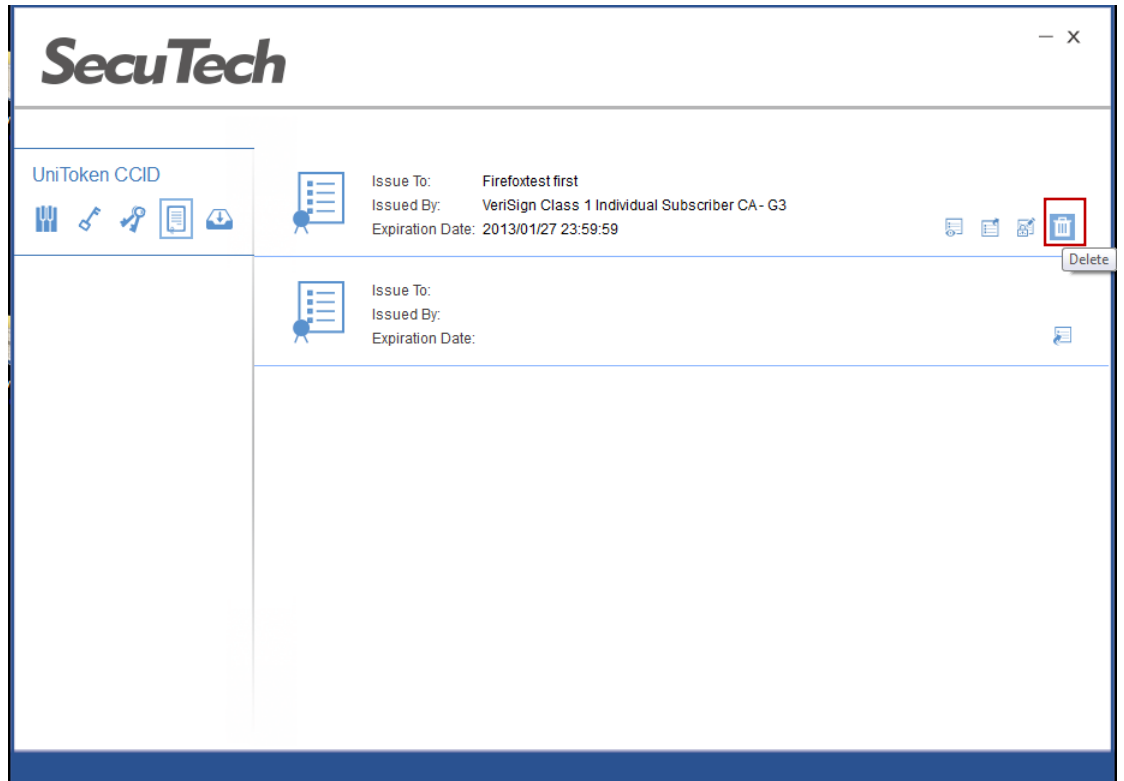


- Verification

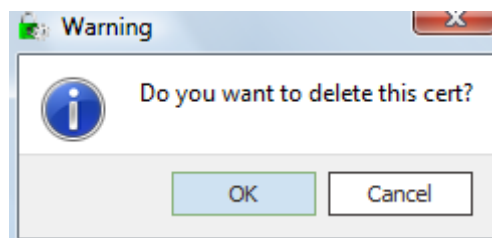
In this page, click on verify to verify a signature signed by this certificate.

- Delete certificate

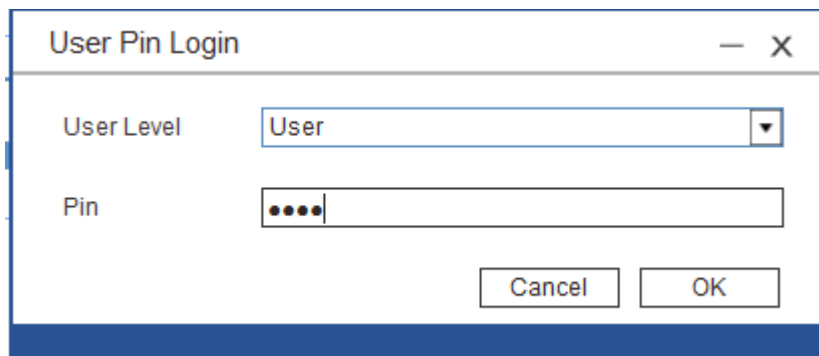
Click on the delete icon



Click on OK



Click on OK and in the pop up page input user PIN.



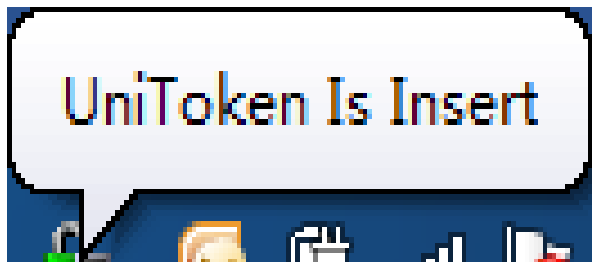
Click on OK to delete the certificate.

## 4.4 Monitor

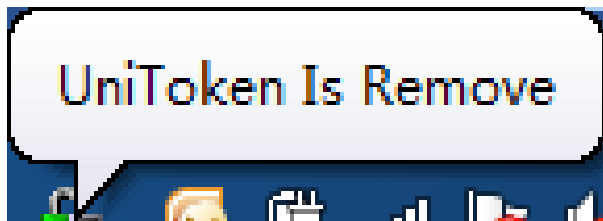
UniToken Monitor is used to view the detailed information of certificates imported into the UniToken and register or unregister certificates. Here, it also provides a way to change User PIN.

### 4.4.1 *Monitor device*

Start UniTokenMonitor.exe and insert token.



Unplug device



### 4.4.2 *Operation*

1. Start Monitor.exe and select a target device.

X

## SecuTech

---

### Equipment Manage

Select Token.

### Certificate Manage

Register the certificate, Import certificate from a file.  
View the certificate details.

Issue To	Issue From	Expire Time	Cert State

Register

UnRegister

View

Import

### Expiration Reminder

Set expiration reminder time.

30

Days

OK

Version 2.3.0

- Change password  
Click on ChangePwd, and in the pop up window input old password and new password.



✕

### Change Password

Change the user password.

Old Pwd:

New Pwd:

Confirm Pwd:

OK

Click on OK

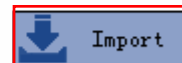
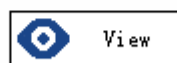
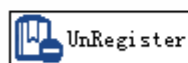
✕

### Prompt Message

Change Password Is Succeed!

- Import certificate  
Click on the import.

Issue From	Expire Time	Cert State

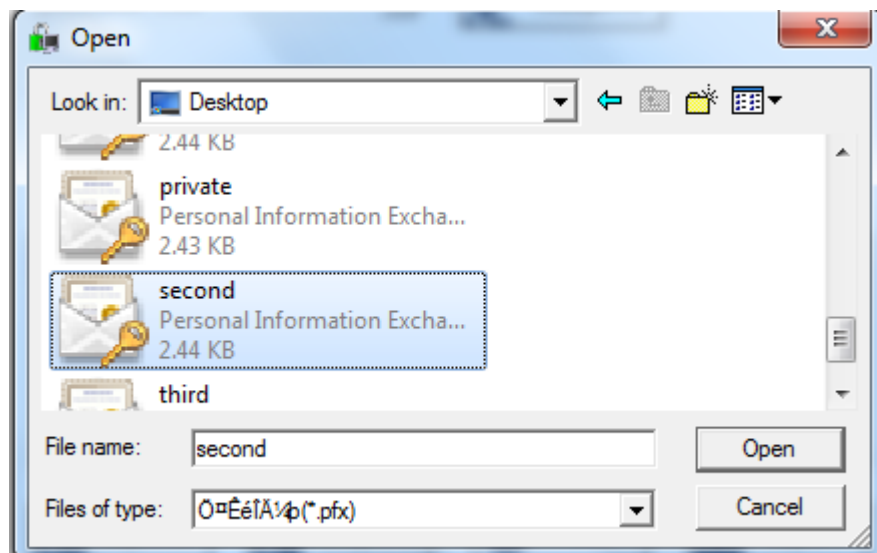


### Reminder

In the pop up page, input user password and click on login.



Find the certificate to be imported.



Enter the password to the certificate.



Click on OK.



Certificate will display in the device.

**SecuTech**

---

**Equipment Manage**  
Select Token.

UniToken CCID

**Certificate Manage**  
Register the certificate, Import certificate from a file.  
View the certificate details.

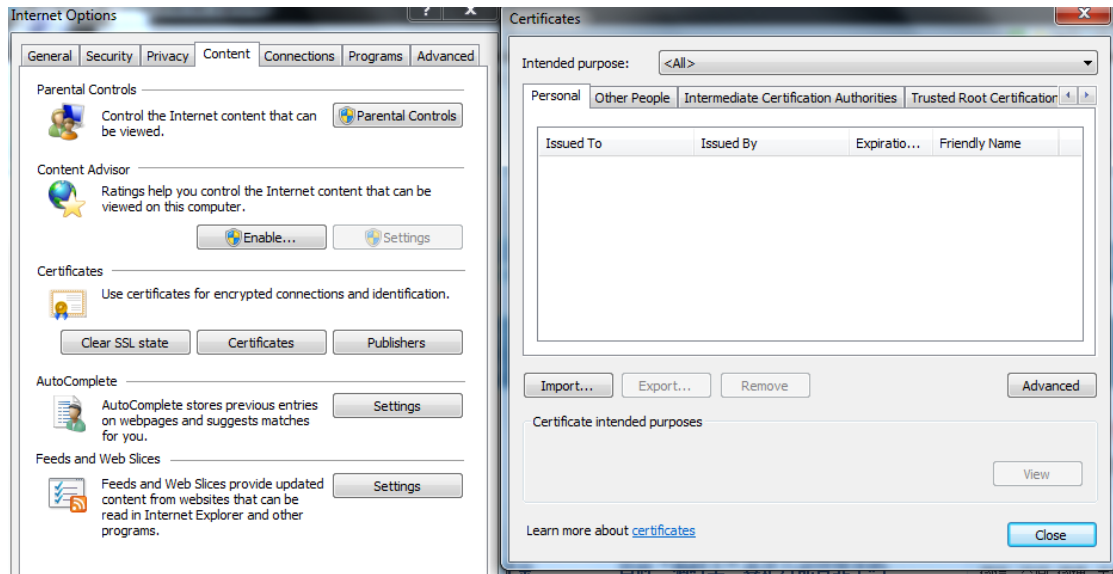
Issue To	Issue From	Expire Time	Cert State
Firxfax second	VeriSign Cla...	2013/01/28 0...	Register Suc...

**Expiration Reminder**  
Set expiration reminder time.

Days

Version 2.3.0

- Register certificate  
In IE-tool-internet options-content-certificates check the registered certificates.  
There is no certificate if it's first used.



Select the certificate to be imported.

**SecuTech**

**Equipment Manage**  
Select Token.

UniToken CCID

**Certificate Manage**  
Register the certificate, Import certificate from a file.  
View the certificate details.

Issue To	Issue From	Expire Time	Cert State
Firxffox second	VeriSign Cla...	2013/01/28 0...	Register Suc...

**Expiration Reminder**  
Set expiration reminder time.

Days

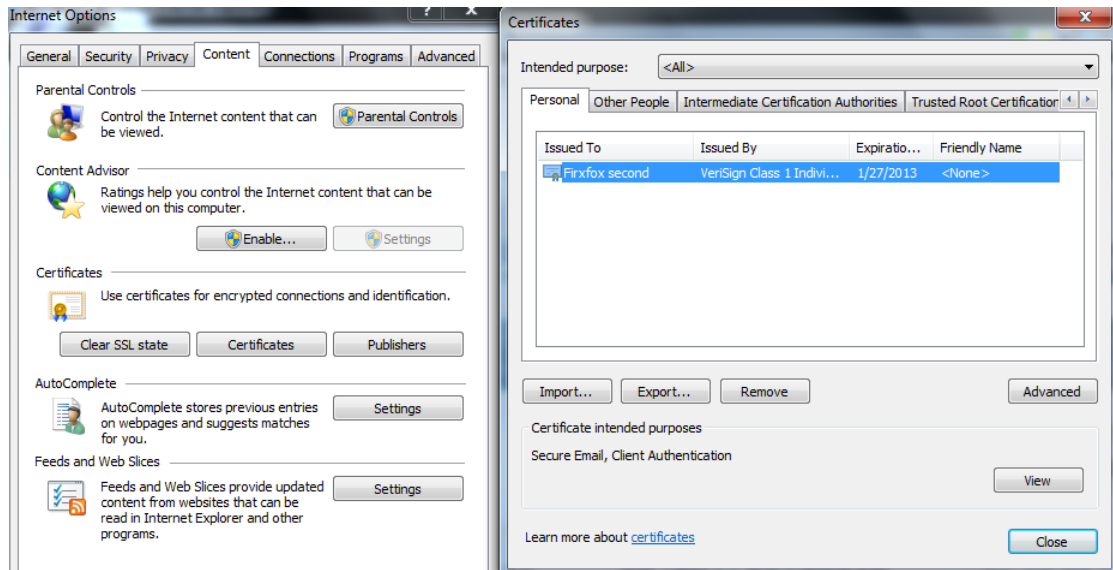
Version 2.3.0

Click on Register

**Prompt Message**

Register Success!

From IE-internet options-content-certificates, the registered certificate can be found.



- Unregister certificate  
Select the certificate to be unregistered.

X

## SecuTech

---

### Equipment Manage

Select Token.

UniToken CCID ▼

ChangePwd

### Certificate Manage

Register the certificate, Import certificate from a file.  
View the certificate details.

Issue To	Issue From	Expire Time	Cert State
Firefox second	VeriSign Cla...	2013/01/28 0...	Register Suc...

Register

UnRegister

View

Import

### Expiration Reminder

Set expiration reminder time.

30

Days

OK

Version 2.3.0

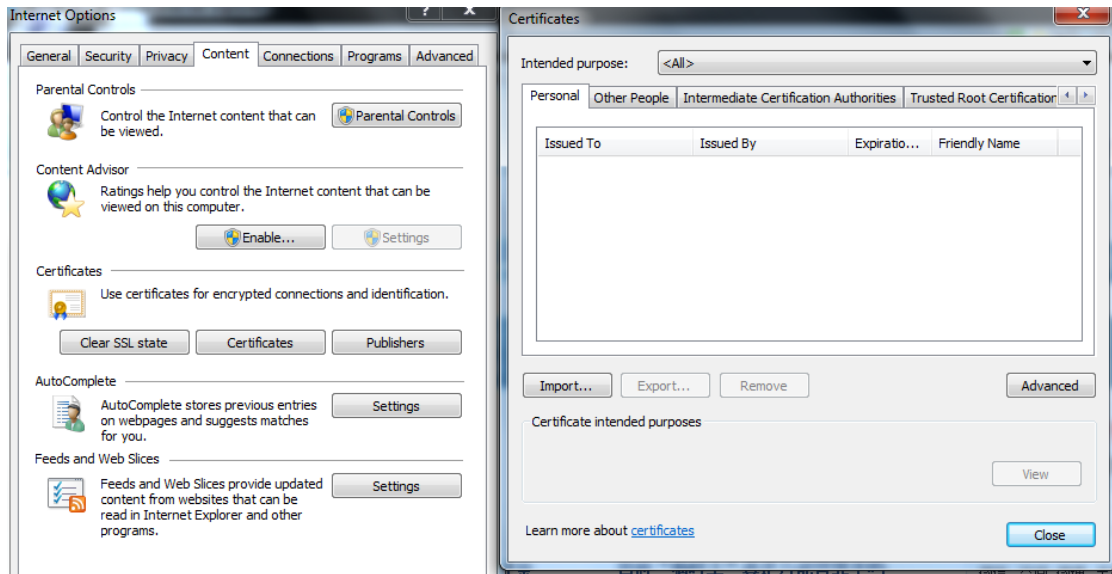
Click on Unregister

X

### Prompt Message

UnRegister Success!

In IE-tool-internet option-content-certificates, the unregistered certificate is removed.



- View certificate information  
Select a certificate



X

## SecuTech

---

### Equipment Manage

Select Token.

▼
 ChangePwd

### Certificate Manage

Register the certificate, Import certificate from a file.  
View the certificate details.

Issue To	Issue From	Expire Time	Cert State
Firefox second	VeriSign Cla...	2013/01/28 0...	Not Register
Firefoxtest first	VeriSign Cla...	2013/01/28 0...	Register Suc.

Register

UnRegister

View

Import

### Expiration Reminder

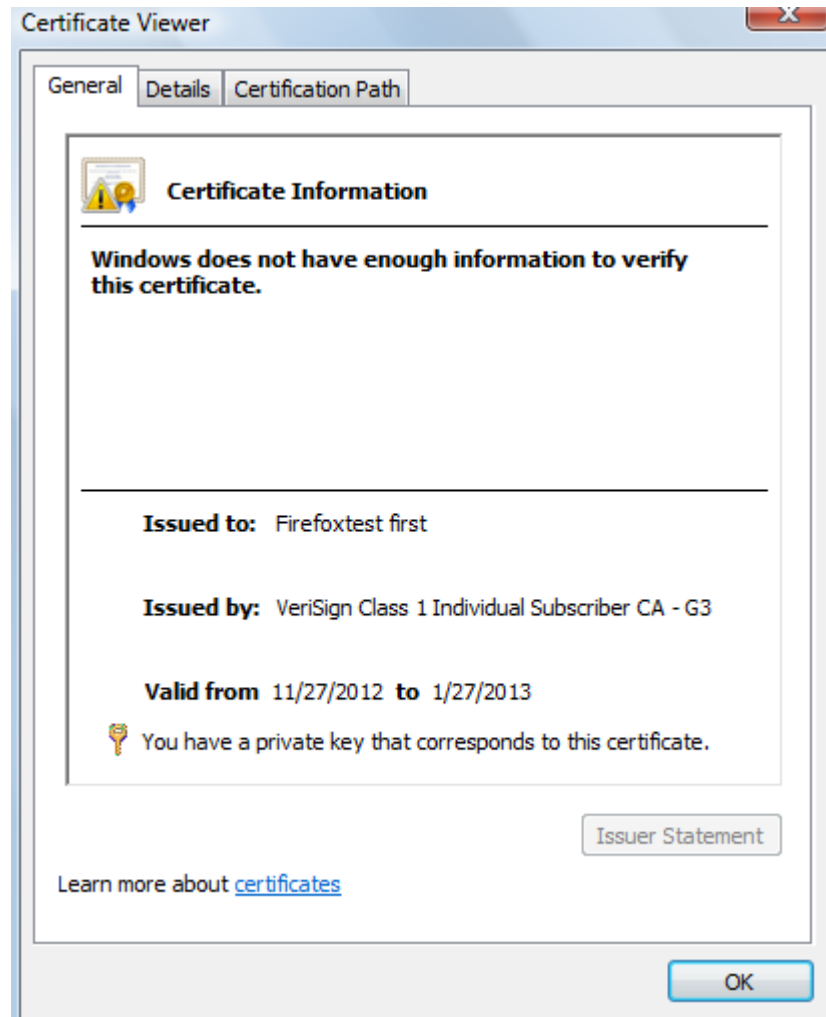
Set expiration reminder time.

Days

OK

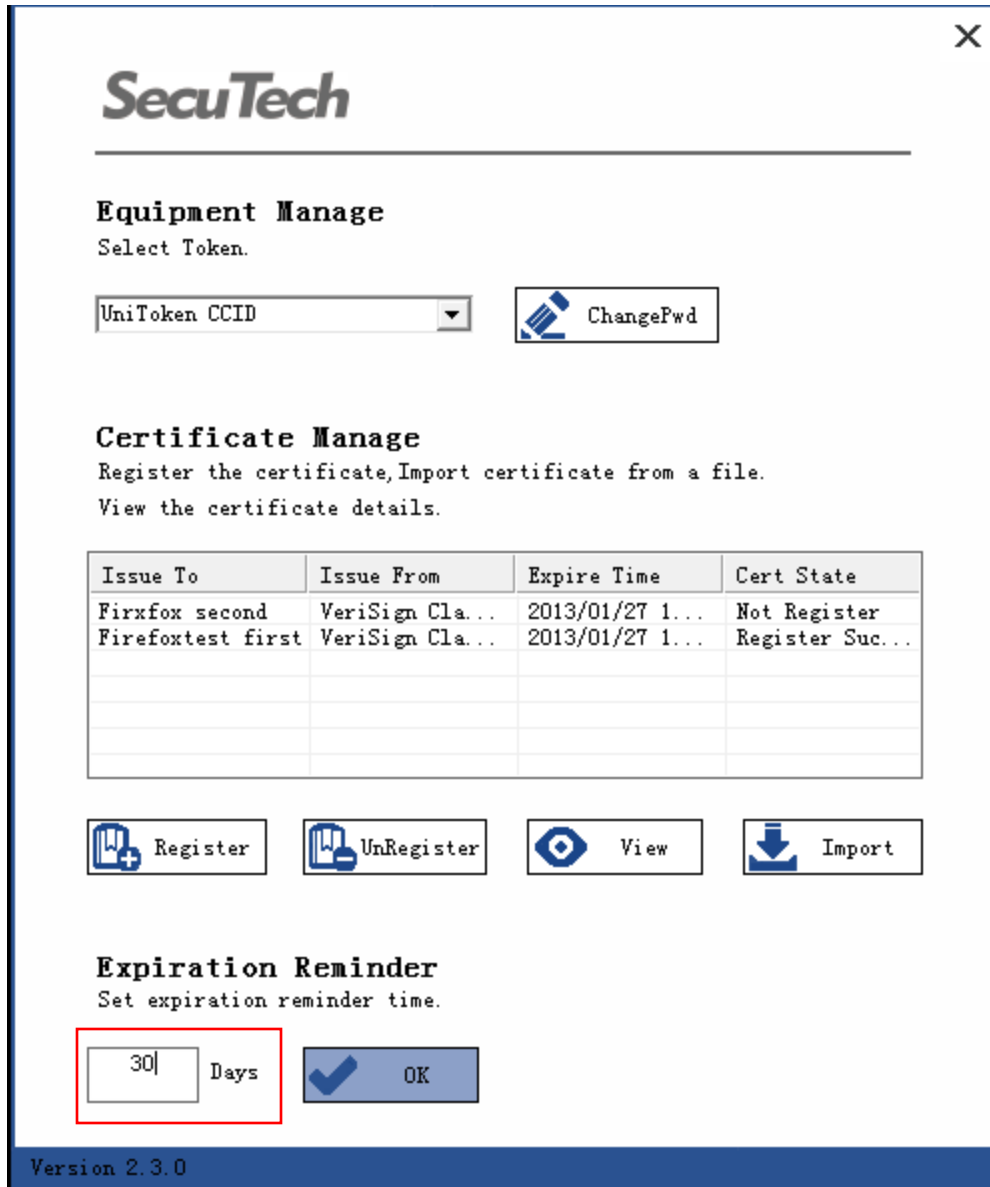
Version 2.3.0

Click on view, and the certificate information will display in the pop up page.

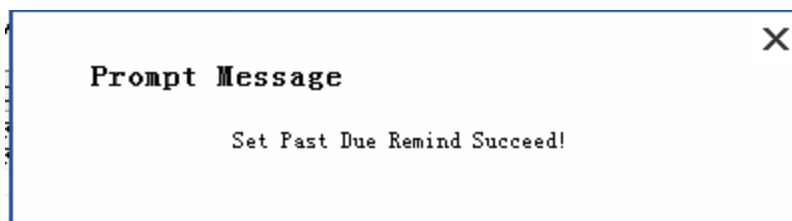


#### 4.4.3 **Expiration reminder**

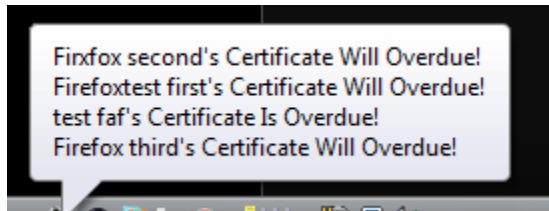
In onitor.exe, input expiration reminder time



Click on OK



If a certificate expire date is less than the reminding date, a reminding message will display shown as the following picture.



## Part 3 Applying Digital Certificates

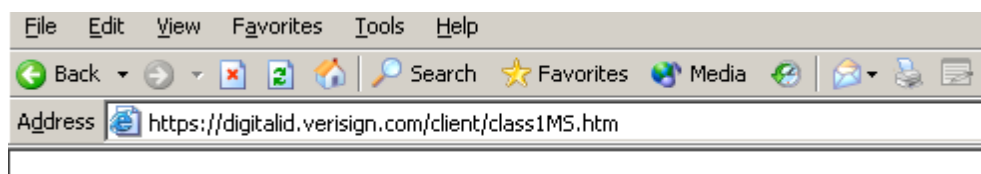
### Chapter 1: Applying Digital Certificates

Token provides a perfect container for digital certificates. Token supports X.509 digital certificates. Token PKI package is the middleware software, which provides digital certificate usage. (See also 1.4.2)

Digital certificate is used to certify that the Token is the right device. Without it, any operation of the Token is forbidden. In this part, we will introduce how to apply digital certificates. We will take the VeriSign certificate and Microsoft Certificate for example.

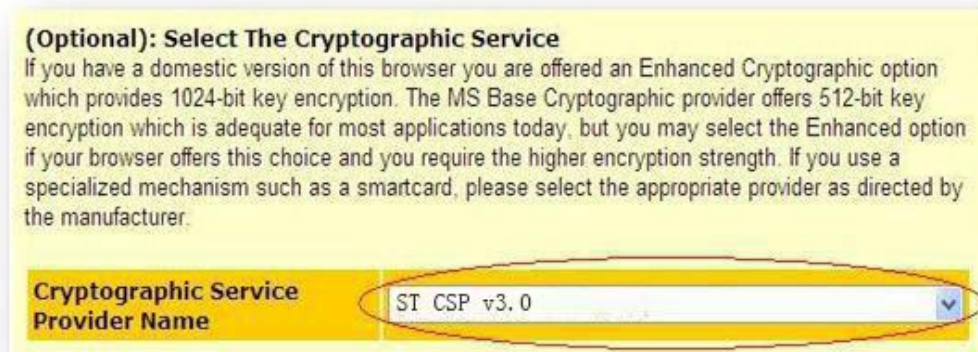
#### 1.1 Applying VeriSign Certificates

Insert one Token into USB port first, and start IE, type in <https://digitalid.verisign.com/client/class1MS.htm> to open the certificate applying page.



There are four steps for applying a certificate. The page provides comprehensible instructions. It is easy to apply certificates by following the instructions step by step.

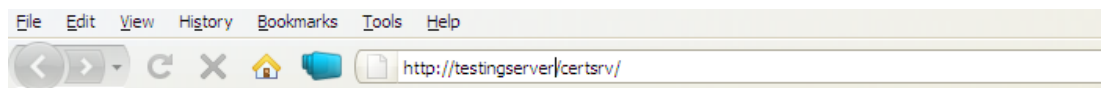
In particular, at the step of complete the enrollment, after filling all the information required, select ST CSP v3.0 from the drop down list of Cryptographic Service Provider Name.



Click "Accept" to start processing. In the following steps, you should check e-mail, pick up digital ID and then install the digital ID according to the page tips. RSA encryption key is generated in the Token.

If more than one Token are inserted in USB ports, please select the Token you want to perform this operation. "Logon" dialog box will pop up and User PIN needs to be input.

## 1.2 Applying Microsoft Certificates



Insert one Token into USB port first, and start IE to open Microsoft certificate applying page.

This is the home page of the certificate applying site. Firstly, you should click Request a certificate.

### Welcome

Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Active Directory Certificate Services, see [Active Directory Certificate Services Documentation](#).

### Select a task:

[Request a certificate](#)

[View the status of a pending certificate request](#)

[Download a CA certificate, certificate chain, or CRL](#)

And then, select advanced certificate request.

### Request a Certificate

Select the certificate type:

[User Certificate](#)

Or, submit an [advanced certificate request](#).

On the page of Advanced Certificate Request, select create and submit a request to this CA.

### Advanced Certificate Request

The policy of the CA determines the types of certificates you can request. Click one of the following options to:

[Create and submit a request to this CA.](#)

[Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-64-encoded PKCS #7 file.](#)

For certificate template, select smartcard logon in the list; for CSP, select ST CSP v3.0

Microsoft Certificate Services -- pscST

### Advanced Certificate Request

**Certificate Template:**

Smartcard Logon

**Key Options:**

Create new key set  Use existing key set

CSP: ST CSP v3.0

Key Usage:  Signature

Key Size: 1024 (Min: 512, Max: 1024, common key sizes: 512, 1024)

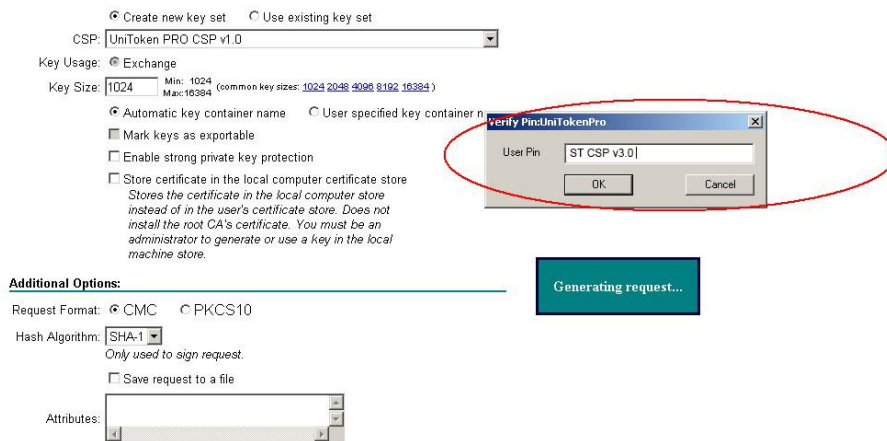
Automatic key container name  User specified key container name

Mark keys as exportable

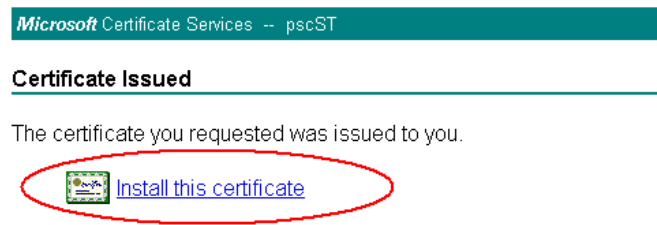
Enable strong private key protection

Store certificate in the local computer certificate store  
*Stores the certificate in the local computer store instead of in the user's certificate store. Does not install the root CA's certificate. You must be an administrator to generate or use a key in the local machine store.*

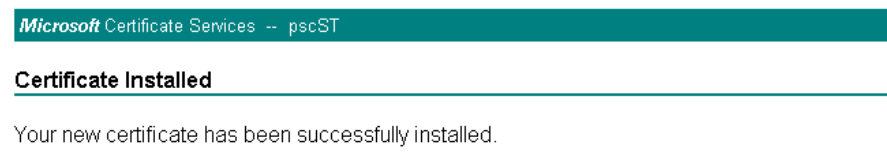
Then, a window will appear to ask you to type in Token PIN. Click "OK" . The system will generate certificate automatically.



Click “install this certificate” for installation.



After installation, the system will prompt that certification has been successfully installed.



## 1.3 Using Digital Certificates

SeuTech provides a series of solutions about the use of digital certificates, in the aspects of IE, Outlook, PDF, Office and so on.

For the detailed instructions about that, please download relative integration guides from [www.eSeuTech.com](http://www.eSeuTech.com).

# Part 4 Developer's Guide

## Device Initialization

Token has been PKI initialization at factory. You can use CCID token in PKI application directly. The default user PIN is "user" and security officer PIN is "admin". To format the PKI application, you can use console in SDK\Utilities\Console\console.exe. To complete the format operation, you need to provide transmission key, which is "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF" by factory default. In real application, we suggest security officer change this key to ensure the device security.

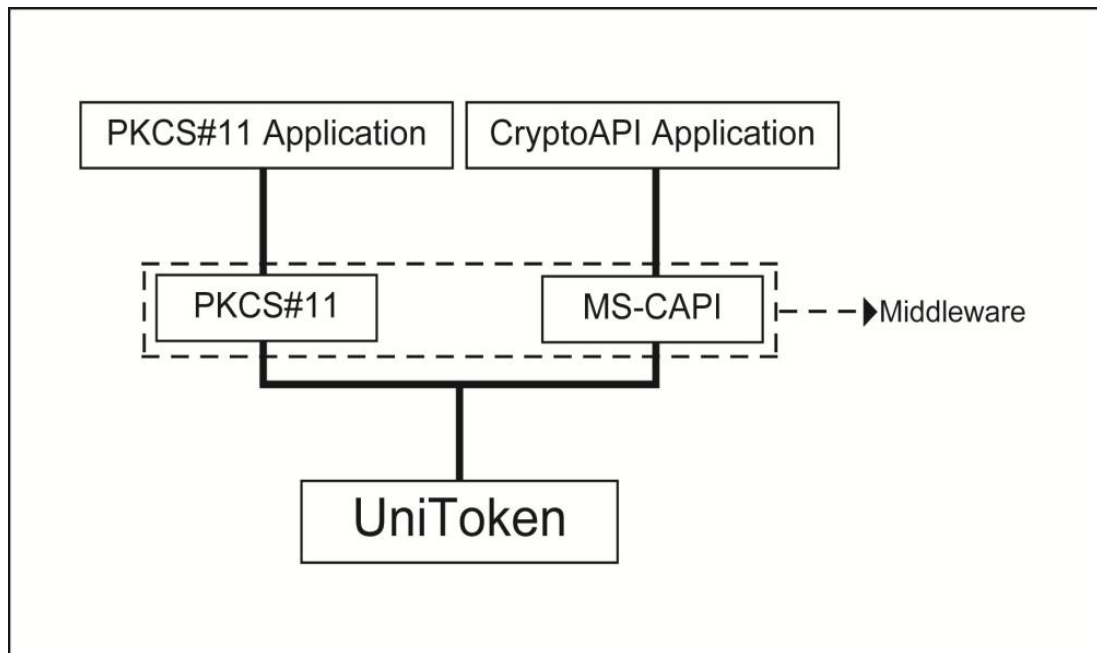
For the third party developers, we provide PKI initialization library and sample which can be found in SDK\Libraries and SDK\Samples respectively.

## Chapter 1: PKCS11 Application

### 1.2 Introduction

PKCS#11 is a Public-Key Cryptography Standard (PKCS) for public key cryptography, developed by RSA Laboratories and includes both algorithm-specific and algorithm-independent implementation standards. It is an industry standard that defines a technology independent programming interface for cryptographic devices such as smartcards and PCMCIA cards. This standard specifies an application program interface (API), called Cryptoki (Cryptographic Token Interface), to devices, either physical or virtual, which hold cryptographic information (keys and other data) and perform cryptographic functions. This API is used across many platforms and is powerful enough for most security-related applications. SecuTech uses PKCS#11 as the main API for Token programming. Token supports PKCS#11 application via Token middleware.





The following files are needed when developing the Token PKCS#11 applications.

Files	Path
Cryptpki.h	Provided by RSA
pkcs11.h	Provided by RSA
Pkcs11f.h	Provided by RSA
Pkcs11t.h	Provided by RSA
uktp11.dll	C:\Windows\system32\

PKCS#11 module of Token supports the creation of the following objects:

Object Class	Description
CKO_DATA	For data structures defined by application
CKO_SECRET_KEY	For symmetric keys
CKO_CERTIFICATE	For X.509 v3 certificates
CKO_PUBLIC_KEY	For RSA/DSA public key
CKO_PRIVATE_KEY	For RSA/DSA private key All

All the objects listed in the above table can be created with Token. The secure storage in Token is limited, so objects can only be created in memory but can NOT be stored in the Token secure storage. Only encryption keys and permanently present data need to be saved in the Token.

### 1.3 Supported PKCS#11 Algorithms and APIs

Mechanisms	Encrypt/Decrypt	Sign/Verify	Digest	Generate key/pair
CKM_RSA_PKCS_KEY_PAIR_GEN				√
CKM_RSA_PKCS				
CKM_DSA_KEY_PAIR_GEN				√
CKM_DSA				
CKM_RC2_KEY_GEN				√
CKM_RC2_ECB				
CKM_RC2_CBC				
CKM_RC2_CBC_PAD				
CKM_RC4_KEY_GEN				√
CKM_RC4				
CKM_DES_KEY_GEN				√
CKM_DES_ECB				
CKM_DES_CBC				
CKM_DES3_KEY_GEN				√
CKM_DES3_ECB				
CKM_DES3_CBC				
CKM_DES3_CBC_PAD				
CKM_MD2				

CKM_MD5				
CKM_SHA_1				
CKM_DH_PKCS_KEY_PAIR_GEN				√
CKM_AES_KEY_GEN				√
CKM_AES_CBC				
CKM_AES_ECB				

The table below lists all the key sizes in Token PKCS#11.

Mechanisms	Key Sizes
CKM_RSA_PKCS_KEY_PAIR_GEN	512~2048bits
CKM_DSA_KEY_PAIR_GEN	512~1024bits
CKM_RC2_KEY_GEN	1~128bits
CKM_RC4_KEY_GEN	1~256bits
CKM_DES_KEY_GEN	8bits
CKM_DES3_KEY_GEN	24bits
CKM_AES_KEY_GEN	16~32bits
CKM_DH_PKCS_KEY_PAIR_GEN	1~128bits

## 1.4 UniMate & UniToken PKCS#11 Function Library

Token PKCS#11 library only implements the standard PKCS #11 APIs. Any other API beyond PKCS#11 is not implemented. If such API is called, an error return code like CKR\_FUNCTION\_NO\_SUPPORT will be returned.

Category	Function	Supported
----------	----------	-----------

General Purpose Function	C_Initialize	YES
	C_Finalize	YES
	C_GetInfo	YES
	C_GetFunctionList	YES
Slot and Token Management Function	C_GetSlotList	YES
	C_GetSlotInfo	YES
	C_GetTokenInfo	YES
	C_WaitForSlotEvent	YES
	C_GetMechanismList	YES
	C_GetMechanismInfo	YES
	C_InitToken	YES
	C_InitPIN	YES
	C_SetPIN	YES
Session Management Function	C_OpenSession	YES
	C_CloseSession	YES
	C_CloseAllSessions	YES
	C_GetSessionInfo	YES
	C_GetOperationState	YES
	C_SetOperationState	YES
	C_Login	YES
	C_Logout	YES
Objects Management Function	C_CreateObject	YES
	C_CopyObject	NO
	C_DestroyObject	YES

	C_GetObjectSize	YES
	C_GetAttributeValue	NO
	C_SetAttributeValue	YES
	C_FindObjectsInit	YES
	C_FindObjects	YES
	C_FindObjectsFinal	YES
Encryption Function	C_EncryptInit	YES
	C_Encrypt	YES
	C_EncryptUpdate	YES
	C_EncryptFinal	YES
Decryption Function	C_DecryptInit	YES
	C_Decrypt	YES
	C_DecryptUpdate	YES
	C_DecryptFinal	YES
Message Digesting Function	C_DigestInit	YES
	C_Digest	YES
	C_DigestUpdate	YES
	C_DigestKey	YES
	C_DigestFinal	YES
Signing and Hashing Function (MAC)	C_SignInit	YES
	C_Sign	YES
	C_SignUpdate	YES
	C_SignFinal	YES
	C_SignRecoverInit	YES

	C_SignRecover	YES
Functions for Verifying Signatures and Hashing (MAC)	C_VerifyInit	YES
	C_Verify	YES
	C_VerifyUpdate	YES
	C_VerifyFinal	YES
	C_VerifyRecoverInit	YES
	C_VerifyRecover	YES
Dual-purpose Cryptographic Function	C_DigestEncryptUpdate	YES
	C_DecryptDigestUpdate	YES
	C_SignEncryptUpdate	YES
	C_DecryptVerifyUpdate	YES
Key Management Function	C_GenerateKey	YES
	C_GenerateKeyPair	YES
	C_WrapKey	NO
	C_UnwrapKey	YES
	C_DeriveKey	NO
Random Number Generation Function	C_SeedRandom	YES
	C_GenerateRandom	YES
Callback Function		YES

## 1.5 Samples

All the samples are implemented in C language, and they all support PKCS#11 standard v.

2.20. For this version, we provide the samples below:

FUNCTION	SAMPLE	DESCRIPTION
To Initialize token	InitToken	The sample is used to initialize token.
To get token information	TokenInfo	The sample is used to get token information.
Encryption/ Decryption	EDcrypt	The sample is used to encrypt and decrypt data.
Sign verification	SignVerify	The sample is used for sign verification.

To initialize token

Path: SDK\sample\PKCS\InitToken\

STEPS	FUNCTION
1. Initialize the PKCS#11 library	C_Initialize
2. Get the slot list	C_GetSlotList
3. Get token information	C_GetTokenInfo
4. Initialize token	C_InitToken
5. Open an session for token	C_OpenSession
6. Log in	C_Login
7. Initialize user PIN	C_InitPIN
8. Log out	C_C_Logout

To get token information

Path: SDK\sample\PKCS\TokenInfo\

STEPS	FUNCTION
1. Initialize the PKCS#11 library	C_Initialize
2. Get the information of PKCS#11 library	C_GetInfo
3. Get the slot list	C_GetSlotList

4. Get the slot information	C_GetSlotInfo
5. Get the token information	C_GetTokenInfo

To verify signature

Path: SDK\sample\PKCS\SignVerify\

STEPS	FUNCTION
1. Initialize the PKCS#11 library	C_Initialize
2. Get the slot list	C_GetSlotList
3. Open an session for token	C_OpenSession
4. Log in	C_C_Login
5. If not found, generate key pair.	C_GenerateKeyPair
6. Initialize a signature	C_SignInit
7. Sign data	C_Sign
8. Initialize verification	C_VerifyInit
9. Verify signature	C_Verify

## Chapter 2: MS-CAPI Applications

### 2.1 Introduction

CAPI (Cryptographic Application Programming Interface), developed by Microsoft as part of Microsoft Windows, is an interface to a library of functions software developers can call upon for security and cryptography services. It is intended for use by developers of applications for MS Windows platforms. CAPI allows multiple cryptographic service providers (CSP) to coexist on the same computer and to be used in the same application. It is also possible to associate a CSP with a particular smartcard, so that smartcard-enabled Windows applications will call the correct CSP. MS Windows contains many helper functions that application developers may use to



simplify code when working with cryptographic functions or with complicated data structures (such as certificates). Choosing which API to use when developing applications is dependent on the needs of the particular application.

## 2.2 Supported Algorithms and APIs

Connection Function	
CPAcquireContext	Create a context and initialize access to CSP which must be specified
CPReleaseContext	Release the context created in CPAcquireContext and other resources
CPGetProvParam	Return information related to CSP
CPSetProvParam	Set parameters of CSP
Key to generate and exchange function	
CPGenKey	Generate key or key pair
CPDeriveKey	Derive a session key from a data hash and guarantee the generated key different
CPSetKeyParam	Set key attribute
CPGetKeyParam	Get the attribute of encryption-operating key
CPExportKey	Export key from container
CPImportKey	Import the key to CSP container
CPDestroyKey	Release key handle, after which the handle will be
	invalid and no access allowed
CPDuplicateKey	Create a duplicate of key
CPGenRandom	Generate random data
CPGetUserKey	Get the enduring key pair from CSP container
Data encryption function	

CPDecrypt	Decrypt encrypted document
CPEncrypt	Encrypt unencrypted document
CPCreateHash	Create hashing objects and initialize them
CPDestroyHash	Delete hashing objects handle
CPDuplicateHash	Create a duplicate of hashing object
CPHashData	Hash the input number
CPGetHashParam	Get the computing result of hashing object
CPHashSessionKey	Hash a session key but no reveal of the key value to application
CPSetHashParam	Set the attribute of a hashing object
CPSignHash	Sign a hashing object
CPVerifySignature	Verify a digital signature

## 2.3 Samples

All the samples are implemented in C language, and they all support MS-CAPI standard. For the standard, we provide the samples below:

Path: SDK\sample\CAPI

FUNCTION	FILES	DESCRIPTION
Algorithm	algorithmTest.cpp algorithmTest.h	The sample provides the operations on symmetric keys, hashing and asymmetric keys.
Container	kcsTest.cpp kcsTest.h	The sample provides the operations on enumeration, delete and creation of files.
Certificates	listcerts.cpp listcerts.h	The sample provides the operations on certificate list.

Algorithm sample

The samples include 3 functions:

```
int GenerateAlgTest(ULONG ulALG); int DeviceAlgTest(ULONG ulALG); int RstTest(ULONG version);
```

GenerateAlgTest is used for DES key generation, encryption and decryption operations.

STEPS	FUNCTION
1. Create a container	CryptAcquireContext
2. Retrieve parameters that govern the operations of a CSP	CryptGetProvParam
3. Generate a key	CryptGenKey
4. Data Encryption	CryptEncrypt
5. Data Decryption	CryptDecrypt

DeviceAlgTest is used for key derivation, data encryption and decryption operations.

STEPS	FUNCTION
1. Create a container	CryptAcquireContext
2. Initiate the hashing of a stream of data	CryptCreateHash
3. Add data to a specified hash object	CryptHashData
4. Derive a key	CryptDeriveKey
4. Data Encryption	CryptEncrypt
5. Data Decryption	CryptDecrypt

RstTest is used for RSA key generation, data encryption and decryption operations.

STEPS	FUNCTION
1. Create a container	CryptAcquireContext

2. Generate a key	CryptGenKey
3. Data Encryption	CryptEncrypt
4. Data Decryption	CryptDecrypt

### Container Sample

The sample demonstrates how to enumerate, add and delete containers with int kcsTest(ULONG ulActive) function.

For enumerating a container

STEPS	FUNCTION
1. Acquire a "VERIFYCONTEXT" handle	CryptAcquireContext
2. Enumerate the key containers	CryptGetProvParam
3. Acquire a handle to the key container found	CryptAcquireContext
4. Try to get a handle to the key pair	CryptGetUserKey
5. Get key permissions	CryptGetKeyParam
6. Display key permissions	

For adding a container

STEPS	FUNCTION
1. Check whether the container already exists	CryptAcquireContext
2. If not, create a container	CryptAcquireContext

For deleting a container

STEPS	FUNCTION
1. Check whether the container already	CryptAcquireContext

exists	
2. If there is, release the handle to the context	CryptReleaseContext
3. Delete the container	CryptAcquireContext

### List Certificate Sample

The sample demonstrates how to enumerate certificates with `int listcerts(void)` function  
 For enumerating certificates

STEPS	FUNCTION
1. Open a handle to the MY\TokenStore certificate store	CertOpenStore
2. Go over each and every certificate within the certificate store	CertEnumCertificatesInStore
3. Get and display the subject name from the certificate	CertGetNameString

## 2.4 UniMate & UniToken API

(See also UniMate & UniToken API Reference in Token SDK\Documents\)



About SecuTech

SecuTech Solution Inc. is a company specializing in data protection and strong authentication, providing total customer satisfaction in security systems & services for banks, financial institutions & other industries. Having extensive and in-depth experience within the information security market, SecuTech has drawn upon this experience to utilize today's cutting-edge technologies, enables enterprises, financial institutions, and government to safely adopt the economic benefits of mobile and cloud computing that are effective against increasingly sophisticated cyber attacks.

**SecuTech** [www.eSecuTech.com](http://www.eSecuTech.com) SecuTech Solution Inc.

North America	China	APAC	EMEA
1250 Boulevard René-Lévesque Ouest, #2200, Montreal, QC, H3B 4W8, Canada T: +1 -888-259-5825 F: +1 -888-259-5825 ext.0 E: <a href="mailto:INFO@eSecuTech.com">INFO@eSecuTech.com</a>	Level 12, #67 Bei Si Huan Xi Lu, Beijing, China, 100080 T: +8610-8288 8834 F: + 8610-8288 8834 E: <a href="mailto:CN@eSecuTech.com">CN@eSecuTech.com</a>	Suite 5.14, 32 Delhi Rd, North Ryde, NSW, 2113, Australia T: 00612-9888 6185 F: 00612-9888 6185 E: <a href="mailto:AUS@eSecuTech.com">AUS@eSecuTech.com</a>	4 Cours Bayard 69002 Lyon, France T: +33-042-600-2810 F: +33-042-600-2810 M: +33-060-939 6463 E: <a href="mailto:Europe@eSecuTech.com">Europe@eSecuTech.com</a>

© Copyright 2012 SecuTech Solution Inc. All rights reserved. Reproduction in whole or in part without written permission from SecuTech is prohibited. SecuTech Token and the SecuTech Logo are trademarks of SecuTech Inc. Windows and all other trademarks are properties of their respective owners. Features and specifications are subject to change without notice.