Elite LC

Clave2 Basic v2.2.2.2

API Reference

SENSELOCK
Reliable Professional Partner

Copyrights and Trademarks

All attempts have been made to make the information in this document complete and accurate. Senselock is not responsible for any direct or indirect damages or loss of business resulting from inaccuracies or omissions. The specifications contained in this document are subject to change without notice.

Contact

SENSELOCK SOFTWARE TECHNOLOGY CO., LTD.
Suite 1706, Culture Square,
Jia 59 ZhongGuanCun Street, Haidian District,
Beijing 100872,
P.R. China

Tel.: +86-10-82642305
Fax: +86-10-51581365

E-mail: info@senselock.com
Website: www.senselock.com

License Agreement

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE CONTENTS THEREOF AND/OR BEFORE DOWNLOADING OR INSTALLING THE SOFTWARE PROGRAM. ALL ORDERS FOR AND USE OF THE Elite AND/OR Clave2 FAMILY PRODUCTS (including but not limited to the Kit, libraries, utilities, diskettes, disc, Senselock® and/or Senselock® keys, the software component of Senselock and/or Clave2 and the Clave2 License Guide) (hereinafter “Product”) SUPPLIED BY Senselock Software Technology Co., Ltd (herein after "Senselock") ARE AND SHALL BE, SUBJECT TO THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT.

This document is a legally binding agreement between you (either an individual or an entity) and Senselock®. If you are not willing to be bound by the terms of this agreement, you should promptly (and at least within 3 days from the date you received this package) return the unused developer's kit and the programmer's guide to Senselock. Use of the software indicates your acceptance of these terms.

- **GRANT OF LICENSE**

  The software of the Product is being licensed to you, which means you have the right to use the software only in accordance with this License Agreement. You may (a) copy the software for internal use, (b) modify the software for the purpose of integrating with your application and (c) merge the software with other programs.

- **NON-PERMITTED USES**

  Except explicitly permitted in this License Agreement, you may not (a) copy, modify, reverse engineering, decompose, assemble the Product in whole or in part, or (b) sell, lease, license, transfer, distribute all or part of the Product or rights granted in this License Agreement.

- **LIMITED WARRANTY**

  After the date of purchase, Senselock provides 12-month warranty that the Senselock Clave2 key has no material and manufacturing defects substantially. All the responsibilities of Senselock and all the compensation you can get under warranty are: you can require replace/repair the Product or accept other remedial measures.

- **LIMITATION OF LIABILITY**

  Under any circumstances, Senselock will NOT be liable for any damages arising out of usage or inability of the Product, including but not limited to: loss of data, loss of profits, and other special, incidental, joint, secondary or indirect loss.

  Except for the limited warranty offered to the original buyer, Senselock is not responsible for providing any insurance to anyone on the product, performance and service including merchantability and fitness for a particular purpose.

The entire product, including Clave2, the software, the document, other material shipped as accessories, and backups made by you are copyrighted by Senselock.

- **TERMINATION**

  Your failure to comply with the terms of this License Agreement shall terminate your license and this License Agreement.

Contents

1

Overview

About the Guide

Mode	Model	Version	Releasing Date
Elite LC	Clave2 Basic	V2.2.2.2	June 2014

CONVENTIONS USED

The following conventions are used throughout this document:

Italic	File Names and Directory Names.
Bold	Keystrokes, Menu Items, and Window Names and Fields
`Consolas`	API parameter
Arial	API Macro, Error Code
CAP	API Struct
⚠	Critical Information

DOCUMENT IMPROVEMENT

Document Writing Team dedicates to insure the accuracy and completeness of context. Your feedback will assist them to make continuous improvement on Clave2 document. Please do not hesitate to email us info@senselock.com.

2

Device API

LC_open

Open a matching device according to a valid Developer ID and index (in case of multiple devices plugged in).

```
int LCAPI LC_open(
    IN int vendor,
    IN int index,
    OUT lc_handle_t *handle
);
```

Parameters

vendor	[IN] Developer ID (0=all)
Index	[IN] Index of devices (0=first, and so on)
handle	[OUT] Returned device handle

Return Value

Device not found	LC_FIND_DEVICE_FAILED
Device found but failed to open	LC_OPEN_DEVICE_FAILED
Device open	LC_SUCCESS handle returns device handle

Remarks

Open in exclusive mode.

LC_close

Close a device.

```
int LCAPI LC_close(
IN lc_handle_t handle
);
```

Parameters

`handle`	[IN] Opened device handle

Return Value

Device closed successfully	LC_SUCCESS
Device failed to close	Go to error code

Remarks

Device cannot be closed repeatedly.

Closing device will also clear out the password validation status.

LC_passwd

Validate a password.

```
int LCAPI LC_passwd(

IN lc_handle_t handle,

IN int type,

IN unsigned char *passwd

);
```

Parameters

`handle`	[IN] Opened device handle
`type`	[IN] Password type (0=Admin, 1=User, 2=Authentication)
`password`	[IN] Password (8 bytes)

Return Value

Invalid password	LC_INVALID_PASSWORD
Valid password	LC_SUCCESS

Remarks

Valid password grants corresponding privileges.

LC_read

Read data from specified block.

```
int LCAPI LC_read(
IN lc_handle_t handle,
IN int block,
OUT unsigned char *buffer
);
```

Parameters

handle	[IN] Opened device handle
block	[IN] Block number (0~3)
buffer	[OUT] Read from data buffer (>=512 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires to verify a password before reading data. (any type password)

LC_write

Write data to specified block.

```
int LCAPI LC_write(
IN lc_handle_t handle,
IN int block,
IN unsigned char *buffer
);
```

Parameters

handle	[IN] Opened device handle
block	[IN] Block number (0~3)
buffer	[IN] Write into data buffer (>=512 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires to verify a password before writing data. (Block 0 is writable to any privileges, Block 1~3 is only writable to admin privileges.)

LC_encrypt

Use internal AES algorithm to encrypt data.

```
int LCAPI LC_encrypt(
IN lc_handle_t handle,
IN unsigned char *plaintext,
OUT unsigned char *ciphertext
);
```

Parameters

handle	[IN] Opened device handle
plaintext	[IN] Plaintext to be encrypted (16 bytes)
ciphertext	[OUT] Ciphertext, after being encrypted (16 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires user privileges.

LC_decrypt

Use internal AES algorithm to decrypt data.

```
int LCAPI LC_decrypt(
IN lc_handle_t handle,
IN unsigned char *ciphertext,
OUT unsigned char *plaintext
);
```

Parameters

`handle`	[IN] Opened device handle
`ciphertext`	[IN] Ciphertext to be decrypted (16 bytes)
`plaintext`	[OUT] Plaintext, after being decrypted (16 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires User privileges.

LC_set_passwd

Set up a new password.

```
int LCAPI LC_set_passwd(
IN lc_handle_t handle,
IN int type,
IN unsigned char *passwd,
IN int retries
);
```

Parameters

handle	[IN] Opened device handle
type	[IN] Password type (0=Admin, 1=User, 2=Authentication)
passwd	[IN] New password (8 bytes)
retries	[IN] Retries counter (1~15), setting -1 disables the function of retries. Retries is not applicable on Admin and User Password.

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires Admin privileges.

LC_change_passwd

Change password.

```
int LCAPI LC_change_passwd(

IN lc_handle_t handle,

IN int type,

IN unsigned char *oldpasswd,

IN unsigned char *newpasswd

);
```

Parameters

`handle`	[IN] Opened device handle
`type`	[IN] Password type
`oldpasswd`	[IN] Old password (8 bytes)
`newpasswd`	[IN] New password (8 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

Only Authentication privileges can change password by using this function. Other users can only use **LC_set_passwd** to reset.

LC_get_hardware_info

Read device information.

```
int LCAPI LC_get_hardware_info(

IN lc_handle_t handle,

OUT LC_hardware_info *info

);
```

Parameters

`handle`	[IN] Opened device handle
`info`	[OUT] Struct pointer storing hardware information

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

LC_get_software_info

Read version information.

```
int LCAPI LC_get_software_info(

OUT LC_software_info *info

);
```

Parameters

`info`	[OUT] Struct pointer storing software information

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

LC_Hmac()

Calculate HMAC value by hardware.

```
int LCAPI LC_hmac (

IN lc_handle_t handle,

IN unsigned char *text,

IN int textlen,

OUT unsigned char *digest

);
```

Parameters

`handle`	[IN] Opened device handle
`text`	[IN] The data that needs to be HMAC value
`textlen`	[IN] Length of text (by byte, >=0)
`digest`	[OUT] HMAC value (20 bytes)

Return Value

Successful	**LC_SUCCESS** `digest` returns the calculated HMAC value
Failed	Go to error code

Remarks

It requires Authentication privileges.

LC_Hmac_software()

Calculate HMAC value by software.

```
int LCAPI LC_hmac_software (

IN unsigned char *text,

IN int textlen,

IN unsigned char *key

OUT unsigned char *digest

);
```

Parameters

`text`	[IN] The data that needs to be HMAC value
`textlen`	[IN] Length of text (by byte, >=0)
`key`	[IN]key of HMAC algorithm (20 bytes Hex string)
`digest`	[OUT, RETVAL] HMAC value (20 bytes)

Return Value

Successful	**LC_SUCCESS** `digest` returns the calculated HMAC value
Failed	Go to error code

Remarks

It requires authentication privileges.

While using the device to authenticate, parameter `key` must be identical to the device Authentication Key.

LC_update

Update remotely.

```
int LCAPI LC_update(

IN lc_handle_t handle,

IN unsigned char *buffer

);
```

Parameters

handle	[IN] Opened device handle
buffer	[IN] Buffer area of remote updating package

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires User privileges.

Must not plug out the device while updating, for it will cause the alteration of Admin Password.

LC_set_key

Reset Remote Update Key and Authentication Key.

```
int LCAPI LC_update(

IN lc_handle_t handle,

IN int type,

IN unsigned char *key,

);
```

Parameters

handle	[IN] Opened device handle
type	[IN] Password type (0=Remote Update Key, 1=Authentication Key)
key	[IN] Key (20 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires Admin privileges.

LC_gen_update_pkg

Generate remote updating package.

```
int LCAPI LC_gen_update_pkg (

IN unsigned char *serial,

IN int block,

IN unsigned char *buffer,

IN unsigned char *key,

OUT unsigned char *uptPkg

);
```

Parameters

serial	[IN] Serial number of updating dongle
block	[IN] Block number (1~3)
buffer	[IN] Updating content (Block in byte)
key	[IN] Remote updating key (in byte)
uptPkg	[OUT] Generated updating package (in byte)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It is operatable without any devices.

Error Code

Decimal Number Returned	Macro	Definition
0	LC_SUCCESS	
1	LC_OPEN_DEVICE_FAILED	
2	LC_FIND_DEVICE_FAILED	
3	LC_INVALID_PARAMETER	
4	LC_INVALID_BLOCK_NUMBER	
5	LC_HARDWARE_COMMUNICATE_ERROR	
6	LC_INVALID_PASSWORD	
7	LC_ACCESS_DENIED	
8	LC_ALREADY_OPENED	
9	LC_ALLOCATE_MEMORY_FAILED	
10	LC_INVALID_UPDATE_PACKAGE	
11	LC_SYN_ERROR	Synchronization error
12	LC_OTHER_ERROR	Unknown error

Script ActiveX Control 3

Brief Description

The Script ActiveX Control is the second encapsulation based on Generic API. This control can be applied on Internet Explorer, and available to develop web pages using this control by VBScript, Javascript and etc.

Sample codes are provided in the SDK.

The Script ActiveX Control covers most of Generic API. Its API definition is as follows:

CLSID: {D9AD0FA7-7515-48B0-87F5-0A9546F9D5E8}
IID: {D9AD0FA7-7515-48B0-87F5-0A9546F9D5E8}
ProgID: LC_SEC.LCSEC.1

Open()

Open a matching device according to its index.

```
HRESULT  Open(
   IN LONG index
   );
```

Parameters

`Index`	[IN] Index of devices (0=first, and so on)

Return Value

Device not found	LC_FIND_DEVICE_FAILED
Device found but failed to open	LC_OPEN_DEVICE_FAILED
Device open	LC_SUCCESS

Remarks

Open in exclusive mode.

Multi-thread not supported.

Close()

Close an open device.

```
HRESULT  Close(
  );
```

Parameters

Return Value

Device closed successfully	LC_SUCCESS
Device failed to close	Go to error code

Remarks

Device cannot be closed repeatedly.

Closing device will also clear out the password validation status.

Multi-thread not supported.

Passwd()

Validate a password.

```
HRESULT  Passwd(
  IN    LONG      type,
  IN    BSTR     passwd
  );
```

Parameters

`type`	[IN] Password type (0=Admin, 2=Authentication)
`password`	[IN] Password(16 bytes)

Return Value

Invalid password	LC_INVALID_PASSWORD
Valid password	LC_SUCCESS

Remarks

Valid password grants corresponding privileges to owner.

Multi-thread supported.

Set_passwd()

Set up a new password.

```
HRESULT  Set_passwd (
  IN    LONG      type,
  IN    BSTR       passwd,
  IN    LONG      retries
  );
```

Parameters

type	[IN] Password type (0=Admin, 2=Authentication)
passwd	[IN] New password (16 bytes)
retries	[IN] Retry counter (1~15), setting -1 disables the function of retries. Retries is not applicable on Admin Password.

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires admin privileges.

Multi-thread supported.

Change_passwd()

Change authentication password.

```
HRESULT  Change_passwd (
   IN    BSTR       oldpasswd,
   IN    BSTR       newpasswd
   );
```

Parameters

`oldpasswd`	[IN] Old password (16 bytes)
`newpasswd`	[IN] New password (16 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

Only authentication privileges can change Authentication password. Other users can only use **Set_passwd()** to reset.

Multi-thread supported.

Get_hardware_info()

Get device information.

```
HRESULT  Get_hardware_info (
  IN                LONG      type,
  OUT,RETVAL    VARIANT    *info
  );
```

Parameters

type	[IN] Type of hardware information (0=Device SN, 1=Production Date)
info	[OUT, RETVAL] hardware information (32 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

Multi-thread supported.

Hmac()

Calculate HMAC value by hardware.

```
HRESULT  Hmac (
  IN               BSTR         text,
  IN               LONG        textlen,
  OUT,RETVAL     VARIANT      *digest
  );
```

Parameters

`text`	[IN] The data that needs the HMAC value
`textlen`	[IN] Length of text (by byte, >=0)
`digest`	[OUT, RETVAL] HMAC value (80 bytes)

Return Value

Successful	LC_SUCCESS `digest` returns the calculated HMAC value
Failed	Go to error code

Remarks

It requires authentication privileges.

Multi-thread supported.

Hmac_software()

Calculate HMAC value by software.

```
HRESULT  Hmac_software (
  IN             BSTR       text,
  IN             LONG      textlen,
  IN             BSTR       key,
  OUT,RETVAL    VARIANT    *digest
  );
```

Parameters

`text`	[IN] The data that needs the HMAC value
`textlen`	[IN] Length of text (by byte, >=0)
`key`	[IN]key of HMAC algorithm (80 bytes Hex string)
`digest`	[OUT, RETVAL] HMAC value (80 bytes)

Return Value

Successful	**LC_SUCCESS** `digest` returns the calculated HMAC value
Failed	Go to error code

Remarks

It requires authentication privileges.

While using the device to authenticate, parameter `key` must be identical to the device Authentication Key.

Multi-thread supported.

Set_key()

Reset remote update key.

```
HRESULT  Set_key (
IN    BSTR       key
  );
```

Parameters

key	[IN] Key (80 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires admin privileges.

Multi-thread supported.

Error Code

It is required to have returned value AND operated with 0xff.

Decimal Number Returned	Macro	Definition
0	LC_SUCCESS	
1	LC_OPEN_DEVICE_FAILED	
2	LC_FIND_DEVICE_FAILED	
3	LC_INVALID_PARAMETER	
5	LC_HARDWARE_COMMUNICATE_ERROR	
6	LC_INVALID_PASSWORD	
7	LC_ACCESS_DENIED	No privileges
8	LC_ALREADY_OPENED	Device is open
9	LC_ALLOCATE_MEMORY_FAILED	
11	LC_SYN_ERROR	Synchronization error
12	LC_OTHER_ERROR	Unknown error
21	LC_ALREADY_CLOSED	Device is closed

Middleware ActiveX Control 4

Brief Description

The Middleware ActiveX Control is the second encapsulation based on Generic API. This control can be applied on any development environment using COM or OLE technology. The supported development platforms are as follows:

- **Microsoft Visual C++**
- **Microsoft Visual Basic**
- **Inprise C++ Builder**
- **Inprise Delphi**

Sample codes are provided in the SDK.

The Middleware ActiveX Control covers most of Generic API. Its API definition is as follows:

CLSID: {EEFB0056-0E92-44A4-8B10-0C0BD75C56A8}
IID: {EEFB0056-0E92-44A4-8B10-0C0BD75C56A8}
ProgID: LC_FULL.LCFULL.1

Open()

Open a matching device according to its index.

```
HRESULT  Open(
  IN LONG index
  );
```

Parameters

`Index`	[IN] Index of devices (0=first, and so on)

Return Value

Device not found	LC_FIND_DEVICE_FAILED
Device found but failed to open	LC_OPEN_DEVICE_FAILED
Device open	LC_SUCCESS

Remarks

Open in exclusive mode.

Multi-thread not supported.

Close()

Close an open device.

```
HRESULT  Close(
  );
```

Parameters

Return Value

Device closed successfully	LC_SUCCESS
Device failed to close	Go to error code

Remarks

Device cannot be closed repeatedly.

Closing device will also clear out the password validation status.

Multi-thread not supported.

Passwd()

Validate a password.

```
HRESULT  Passwd(
  IN    LONG      type,
  IN    BYTE*     passwd
  );
```

Parameters

`type`	[IN] Password type (0=Admin, 2=Authentication)
`password`	[IN] Password(8 bytes)

Return Value

Invalid password	LC_INVALID_PASSWORD
Valid password	LC_SUCCESS

Remarks

Valid password grants corresponding privileges to owner.

Multi-thread supported.

Set_passwd()

Set up a new password.

```
HRESULT  Set_passwd (
  IN    LONG      type,
  IN    BYTE*       passwd,
  IN    LONG      retries
  );
```

Parameters

`type`	[IN] Password type (0=Admin, 2=Authentication)
`passwd`	[IN] New password (8 bytes)
`retries`	[IN] Retry counter (1~15), setting -1 disables the function of retries. Retries is not applicable on Admin Password.

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires admin privileges.

Multi-thread supported.

Change_passwd()

Change authentication password.

```
HRESULT  Change_passwd (
   IN    BYTE*      oldpasswd,
   IN    BYTE*      newpasswd
   );
```

Parameters

`oldpasswd`	[IN] Old password (8 bytes)
`newpasswd`	[IN] New password (8 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

Only authentication privileges can change Authentication password. Other users can only use **Set_passwd()** to reset.

Multi-thread supported.

Get_hardware_info()

Gain device information.

```
HRESULT  Get_hardware_info (
  IN        LONG       type,
  OUT       BYTE*      info
  );
```

Parameters

type

[IN] Type of hardware information (0=Device SN, 1=Production Date)

info

[OUT] hardware information (8 bytes)

Return Value

Successful

LC_SUCCESS

Failed

Go to error code

Remarks

Multi-thread supported.

Hmac()

Calculate HMAC value by hardware.

```
HRESULT  Hmac (
  IN              BYTE*       text,
  IN              LONG        textlen,
  OUT             BYTE*       digest
  );
```

Parameters

text	[IN] The data that needs the HMAC value
textlen	[IN] Length of text (by byte, >=0)
digest	[OUT, RETVAL] HMAC value (20 bytes)

Return Value

Successful	LC_SUCCESS digest returns the calculated HMAC value
Failed	Go to error code

Remarks

It requires authentication privileges.

Multi-thread supported.

Hmac_software()

Calculate HMAC value by software.

```
HRESULT  Hmac_software (
  IN              BYTE*       text,
  IN              LONG      textlen,
  IN              BYTE*       key,
  OUT             BYTE*    *digest
  );
```

Parameters

`text`	[IN] The data that needs the HMAC value
`textlen`	[IN] Length of text (by byte, >=0)
`key`	[IN]key of HMAC algorithm (20 bytes Hex string)
`digest`	[OUT, RETVAL] HMAC value (20 bytes)

Return Value

Successful	LC_SUCCESS `digest` returns the calculated HMAC value
Failed	Go to error code

Remarks

It requires authentication privileges.

While using the device to authenticate, `key` must be identical to the device Authentication Key.

Multi-thread supported.

Set_key()

Reset remote update key.

```
HRESULT  Set_key (

IN    BYTE*      key

  );
```

Parameters

key	[IN] Key (20 bytes)

Return Value

Successful	LC_SUCCESS
Failed	Go to error code

Remarks

It requires admin privileges.

Multi-thread supported.

Error Code

It is required to have returned value AND operated with 0xff.

Decimal Number Returned	Macro	Definition
0	LC_SUCCESS	
1	LC_OPEN_DEVICE_FAILED	
2	LC_FIND_DEVICE_FAILED	
3	LC_INVALID_PARAMETER	
5	LC_HARDWARE_COMMUNICATE_ERROR	
6	LC_INVALID_PASSWORD	
7	LC_ACCESS_DENIED	No privileges
8	LC_ALREADY_OPENED	Device is open
9	LC_ALLOCATE_MEMORY_FAILED	
11	LC_SYN_ERROR	Synchronization error
12	LC_OTHER_ERROR	Unknown error
21	LC_ALREADY_CLOSED	Device is closed