TEAleaf-USB™

Micro-Sized, Ultra-Low Cost USB Authentication Key for Software Licensing

Summary

TEAleaf-USB is an ultra-low cost, micro-sized authentication system for verifying that a software product is authorized and not a pirate copy. It uses a simple but robust xTEA algorithm to verify that a TEAleaf-USB device is present.

TEAleaf-USB uses the Human Interface Device (HID) USB profile. It does not require USB drivers and is immediately plug-and-play compatible with present and future Windows, Linux and Mac operating systems.

TEAleaf-USB is available as complete, finished product, and also as an integrated circuit and design blueprint for customers with electronics manufacturing capacity. (See *www.firmwarefactory.com.*)

Applications

- Computer software copy protection and licensing
- Pay-per-use hardware protection
- True random number generation

Security

- Random-hashed USB communications believed to be uncrackable
- Quantum limited true random number generator
- Extended Tiny Encryption Algorithm (xTEA)
- 128 bit security key

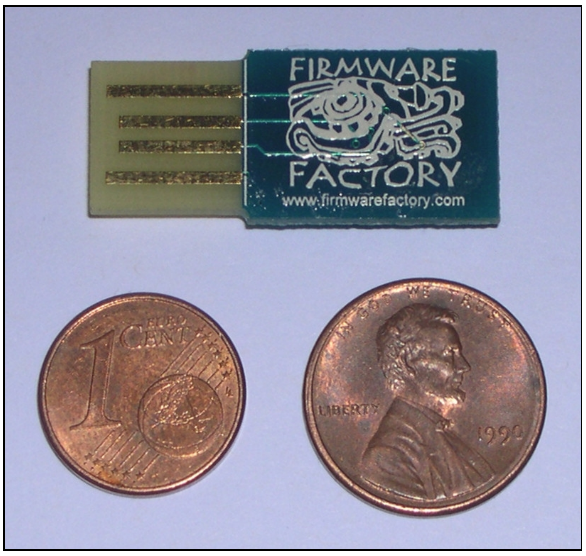
Features

- True USB 2.0 HID plug and play - No drivers required
- Ultra low cost integrated connector
- Security key, product name, manufacturer name, GUID configurable from USB interface
- 122-byte EEPROM

Authentication Mechanism

Authentication requires the PC and the TEAleaf-USB to generate random numbers and then test that they both know how to encrypt and decrypt them. First the PC needs to prove to the TEAleaf that it knows the 128-bit key, then the TEAleaf-USB proves it knows it too. Bits are masked from each proof to ensure that decryption attacks are fruitless.

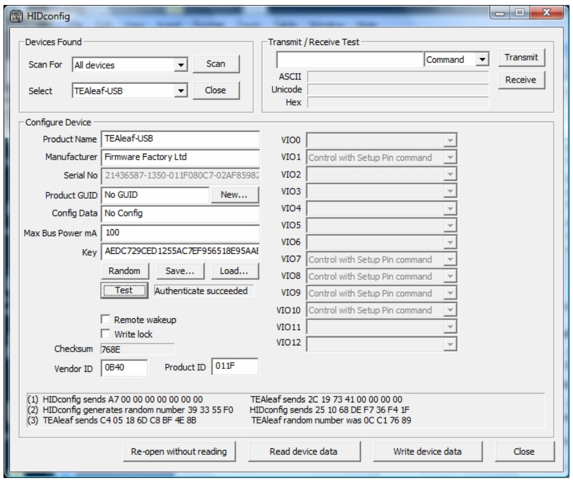
A true random number generator provides the required robustness against hacking. It detects marginal differences in temperature, operating voltage, device chemistry, age and quantum effects to derive a unpredictable value. 192 random bits are generated during startup. These are then fed into a high-avalanche polynomial ring to derive a 192-bit random number.



Device Evaluation

The *HIDconfig.exe* application in the development kit is used for evaluation and customization.

1. Start the application and insert the TEAleaf-USB into a USB socket.
2. Press the *Scan* key. The TEAleaf-USB device should appear in the *Select* list. Press *Open*.
3. You have connected to the TEAleaf-USB. Press *Random…* to generate a random key, then *Save…* to save the key onto your computer.
4. Press *Write device data* to set the TEAleaf key to the random value. The TEAleaf will reset, so you need to press *Open* again.
5. Press *Load* to load the key you previously saved, then *Test* to see if the key is correct. You will see the text *Authenticate Succeeded.*
6. Press *Random* to generate any other key, then *Test* to see if the key is correct. You will see the text *Authenticate Failure.*



Scan For: All devices
Select: TEAleaf-USB
Command
ASCII
Unicode
Hex

Product Name: TEAleaf-USB
Manufacturer: Firmware Factory Ltd
Serial No: 21436587-1350-011F080C7-02AF85982
Product GUID: No GUID
Config Data: No Config
Max Bus Power mA: 100
Key: AEDC729CED1255AC7EF956518E95AAE
Authenticate succeeded
Remote wakeup ☐
Write lock ☐
Checksum: 768E
Vendor ID: 0B40
Product ID: 011F

VIO0
VIO1: Control with Setup Pin command
VIO2
VIO3
VIO4
VIO5
VIO6
VIO7: Control with Setup Pin command
VIO8: Control with Setup Pin command
VIO9: Control with Setup Pin command
VIO10: Control with Setup Pin command
VIO11
VIO12

```
(1) HIDconfig sends A7 00 00 00 00 00 00 00          TEAleaf sends 2C 19 73 41 00 00 00 00
(2) HIDconfig generates random number 39 33 55 F0    HIDconfig sends 25 10 68 DE F7 36 F4 1F
(3) TEAleaf sends C4 05 18 6D C8 BF 4E 8B            TEAleaf random number was 0C C1 76 89
```

Device Configuration

The following non-volatile device configuration settings can be selected from HIDconfig.exe.

Authentication Key

The 128-bit authentication key, expressed as a 32-digit hexadecimal number. This secret key is known only by the *TEAleaf-USB* and your application software. *HIDconfig.exe* can set this value, but not read it.

Product Name / Manufacturer Name

The product / manufacturer names are Unicode strings of up to 61 characters plus zero terminator. The host application can read these data using a Get Feature request for string 1 / 2. The PC displays them when TEAleaf-USB is first inserted. The default values are "TEAleaf-USB" / "Firmware Factory Ltd".

Custom VID / PID

Personalized Vendor and Product IDs are not required, but you can them if you wish. The default Vendor ID is 0x0B40, and the default Product ID is 0x011F.

Write Lock

Once write lock set, no further configuration is possible.

Product GUID

The product GUID is a Unicode string of up to 61 characters plus zero terminator. The host application can read this data using a Get Feature request for string 4. The product GUID is a string which you can use to make a hardware security key specific to a particular software application. It is used to differentiate it from other security keys with the TEAleaf-USB Vendor ID / Product ID combination. It should be the same for all products of the same type. The default value is "No GUID".

Config (EEPROM) String

The configuration data is a Unicode string of up to 61 (122 bytes) characters plus zero terminator. You can use it as you wish to store configuration data on the product which the host software can access. The host application can read this data using a Get Feature request for string 5. The default value is "No Config".

Max Bus Power; Remote Wakeup

These settings should not be modified from the default 100mA / unchecked.

Host-Side Interfacing

TEAleaf-USB uses the Human Interface Device (HID) USB interface. It has the advantages that no device drivers are required, and that a host application can easily locate the TEAleaf.

All exchanges of data ('reports') between the host and the TEAleaf-USB are 8 bytes in length, regardless of how many bytes of meaningful data are actually transferred. In HID terms, all transfers are 10ms interrupt reports of 8 bytes, to and from output ID 0 on EP1.

The host software has two perform two tasks. First it has to locate the device. Then it has to communicate with it. To locate the device, enumerate all devices with Vendor ID 0x0B40 and Product ID of 0x011F. Then use a Get Feature request for the string 4, the Product GUID. If this matches the product GUID you configured for the device, you have located it.

Once you have located the device, open a file to communicate with it. Then send data and receive data as 8-byte reports.

Sample source code for Windows and a Windows dynamic link library (DLL) are provided in the development kit. For a detailed description, please refer to the comments embedded in the source code and the Visual Basic example in the Excel spreadsheet. Sample source code for Mac OS and Linux is in preparation.

Commands

The first byte ('identifier') of the 8-byte report in either direction identifies the remaining contents ('payload'). If the command does not require all 8 bytes, then the contents of the rest are ignored.

Note: Sending a command in the range 0x80-0x8F can modify settings that may permanently disable the device. Other than for R&D purposes, the device should be write-locked when shipped to end-users.

Authenticate

The identifier AUTH (0xA7) is used to initiate the authentication process. The TEAleaf-USB will reply with a four-byte random value in the first 4 bytes. To this the host should append its own four-byte random value in the second 4 bytes and then encrypt using the xTEA algorithm given below. The resulting 8-byte value should be sent to the TEAleaf-USB. It will decrypt the data to determine the host's random value and to verify the random value it sent.

If the value is not correct, TEAleaf will respond with eight zero bytes. If it is correct, it replaces the random number with another random value, encrypts and sends the result to the host. The host decrypts the result to verify the random value it sent to the TEAleaf-USB. If the random value is correct, authentication is complete.

Example:

(Key is the factory default FFEEDDCCBBAA99887766554433221100)

A7 00 00 00 00 00 00 00	Auth
13 16 3A 03 00 00 00 00	Random # from TEAleaf-USB
13 16 3A 03 9B 67 2B 99	Host adds random #
44 92 D0 06 8C F5 90 F2	Host encrypts, sends to TEAleaf-USB
13 16 3A 03 9B 67 2B 99	TEAleaf-USB decrypts, verifies random#
20 2D 6A 18 9B 67 2B 99	TEAleaf-USB adds new random #
45 60 6C 84 BF 86 EE C2	TEAleaf-USB encrypts, sends to host
20 2D 6A 18 9B 67 2B 99	Host decrypts and verifies random #

Random

The identifier RANDOM (0xA8) is used to obtain a 4-byte random number from TEAleaf.

Example:

A8	Random
2B 73 CE 89 00 00 00 00 00	Random # response from TEAleaf

Get Firmware ID

The identifier GETFWID (0x94) retrieves a zero-terminated ASCII text string identifying the firmware and its version number. It will probably need to do so over several response packets.

Example:

94	Command – Get Firmware ID
94 54 45 41 6C 65 61 66	"TEAleaf"
94 2D 55 53 42 20 30 31	"-USB 01"
94 2E 30 30 20 28 32 34	".00 (24"
94 35 30 29 00 91 D5 7E	"50)"

xTEA Algorithm

The xTEA algorithm is a robust Feistel network proposed by Needham & Wheeler. For details refer to *RM Needham and DJ Wheeler, TEA extensions, Technical report, Computer Laboratory, University of Cambridge, October 1997.*

The host-side algorithm is presented below as C code:

```
// unsigned long integers are 32-bit
// unsigned char integers are 8-bit
// Arg pVal is a 2-long array containing your
//   randomly generated challenge
//   pVal[0] is challenge bits C31-C0
//   pVal[1] is challenge bits C63-C0
// Arg pKey is a 4-long array containing key
//   pKey[0] is the least significant 32 bits
//   pKey[1] is the next least significant 32 bits
//   pKey[2] is the next most significant 32 bits
//   pKey[3] is the most significant 32 bits

void Encr(unsigned long *pVal, unsigned long * pKey)
{
  unsigned long sum = 0;
  unsigned long delta = 0x9E3779B9;
  unsigned char i;

  for (i=0; i<32; i++)
  {
    unsigned sa = sum & 0x03;
    unsigned long Key2;
    Key2 = pKey[sa];
    pVal[0] += (( (pVal[1] << 4) ^ (pVal[1] >> 5)) +
                   pVal[1] ) ^ (sum + Key2);
    sum += delta;
    sa =  (sum>>11) & 0x03;
    Key2 = pKey[sa];
    pVal[1] += (( (pVal[0] << 4) ^ (pVal[0] >> 5)) +
                   pVal[0] ) ^ (sum + Key2);
  }
}

// On exit the TEAleaf's response must match pVal
//   pVal[0] is response bits R31-R0
//   pVal[1] is response bits R47-R32
//   ( ignoring 16 highest bits of pVal[1] )
```

The application *HIDconfig.exe* in the development kit pack can generate data for verifying implementations of the algorithm.

Security Considerations

Robustness is a matter of pride and competition between authentication key providers. We therefore are proud to present an key mechanism that is believed to be uncrackable.

The random-hashed xTEA algorithm has a very high avalanche effect and is extremely robust against plaintext and related-key differential attacks. Hashing with random number values generated by both PC and TEAleaf is non-reversible and stops to multiple attacks.

Care should be taken that the host-side algorithm executable code does not expose the key. The following precautions are recommended:

- Deriving the key algorithmically rather than storing explicitly in program code.
- Optimize your code for minimum code size and memory space. This is the hardest to type of code to reverse engineer.
- Insert dummy calculations into the algorithm to make it harder to identify.

Development Kit

A development kit is available for download from *www.flexipanel.com* containing the following files:

- *HIDconfig.exe*, an application which allows you to customize TEAleaf-USB devices via the USB port. It is designed for low labor in-factory use and also serves to test the USB circuit.
- *usb-win.c* and *usb-win.h*, sample HID code for Windows. Additionally the files *setupapi.h*, *hidsdi.h*, *hidpi.h*, *setupapi.lib* and *hid.lib* are provided, which must be included in the application.
- *FwFhid.dll* dynamic link library and Visual Basic example *FwFhidDLLExample.xls*.
- *USB 2.0 Specification* (© HP / Intel / Lucent / Microsoft / NEC / Philips 2000)

Design Blueprint

A design blueprint is available for customers who wish to manufacture TEAleaf-USB. The blueprint requires a signed nondisclosure agreement and consists of the following files:

- *TLUrx BOM.xls* blueprint bill of materials.
- *TLUrx Gerber.zip* blueprint PCB Gerbers.
- *TLUrx BOM.pdf* schematic and component placement diagram.

FlexiPanel™

FlexiPanel Ltd
3 Plough Yard, Gnd Floor
London W1F 9BB, UK
www.flexipanel.com
email: support@flexipanel.com

Manufactured to RoHS,
WEEE & ISO9001:2001 standards

FM76316