Annex A

Feitian Technologies Co., Ltd.

Software Developer's Agreement

All Products of Feitian Technologies Co., Ltd. (Feitian) including, but not limited to, evaluation copies, diskettes, CD-ROMs, hardware and documentation, and all future orders, are subject to the terms of this Agreement. If you do not agree with the terms herein, please return the evaluation package to us, postage and insurance prepaid, within seven days of their receipt, and we will reimburse you the cost of the Product, less freight and reasonable handling charges.

1. **Allowable Use** – You may merge and link the Software with other programs for the sole purpose of protecting those programs in accordance with the usage described in the Developer's Guide. You may make archival copies of the Software.
2. **Prohibited Use** – The Software or ROCKEY2 hardware dongle or any other part of the Product may not be copied, reengineered, disassembled, decompiled, revised, enhanced or otherwise modified, except as specifically allowed in item 1. You may not reverse engineer the Software or any part of the product or attempt to discover the Software's source code. You may not use the magnetic or optical media included with the Product for the purposes of transferring or storing data that was not either an original part of the Product, or a Feitian provided enhancement or upgrade to the Product.
3. **Warranty** – Feitian warrants that the ROCKEY2 dongles and Software storage media are substantially free from significant defects of workmanship or materials for a time period of twelve (12) months from the date of delivery of the Product to you.
4. **Breach of Warranty** – In the event of breach of this warranty, Feitian's sole obligation is to replace or repair, at the discretion of Feitian, any Product free of charge. Any replaced Product becomes the property of Feitian.

Warranty claims must be made in writing to Feitian during the warranty period and within fourteen (14) days after the observation of the defect. All warranty claims must be accompanied by evidence of the defect that is deemed satisfactory by Feitian. Any Products that you return to Feitian, or a Feitian authorized distributor, must be sent with freight and insurance prepaid.

EXCEPT AS STATED ABOVE, THERE IS NO OTHER WARRANTY OR REPRESENTATION OF THE PRODUCT, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

5. **Limitation of Feitian's Liability** – Feitian's entire liability to you or any other party for any cause whatsoever, whether in contract or in tort, including negligence, shall not exceed the price you paid for the unit of the Product that caused the damages or are the subject of, or indirectly related to the cause of action. In no event shall Feitian be liable for any damages caused by your failure to meet your obligations, nor for any loss of data, profit or savings, or any other consequential and incidental damages, even if Feitian has been advised of the possibility of damages, or for any claim by you based on any third-party claim.
6. **Termination** – This Agreement shall terminate if you fail to comply with the terms herein. Items 2, 3, 4 and 5 shall survive any termination of this Agreement.

Table of Contents

Chapter1 Brief Introduction

ROCKEY2 is a traditional secure storage dongle that is extremely simple to implement and low cost. It does not offer many of the advanced software protection methods available in ROCKEY4 or ROCKEY5.

The Features of ROCKEY2 include:

- ROCKEY2 has 2560 bytes of Read/Write memory spaces ,which is much bigger than most of other software protection dongles.
- ROCKEY2 is an HID device so there is no driver required for supported Linux or Windows platforms.
- Each ROCKEY2 has a globally unique hardware ID.
- Multiple ROCKEY2s can work together on the same computer.
- ROCKEY2 supports USB only.

Chapter 2 ROCKEY2 Features

User ID (UID) and Hardware ID (HID)

Each ROCKEY2 dongle contains a User ID (UID) as well as the globally unique Hardware ID (HID). Both the UID and HID are defined as 32-bit DWORD. The UID and HID are identifiers for the ROCKEY2 dongle.

The default UID is "0". While the UID remains at default you may open and read/write to the dongle memory. But you cannot obtain the HID. The HID is shielded while the UID is set to the default. The shielded HID will return a value of "0". The correct HID may be obtained only after the developer resets the UID.

When generating the UID, a 64 bytes string is required as seed code. The UID will be derived from this seed code. The UID generation is performed within the dongle hardware. The process is absolutely irreversible, which means only the user generated the UID knows which seed code is his UID derived from. Even hacker knows UID and knows how to use UID generating program, as he do not know the seed code, he can not reproduce the dongle with the same UID.

Additionally, software vendor can determine whether the store zone is writable or not when generating the UID. If the dongle is set to unwritable, the entire memory zone will be written protected.

ROCKEY2 Driver

The ROCKEY2 dongle is a USB device that uses the driver native to the Windows and Linux operating systems. When the dongle is inserted into the USB port, the operating system will prompt you for installing the new device. With Windows 98 you may need the Win98 installation CD-ROM. You will not need a CD-ROM or other storage media for Win Me/2000/XP – the driver will install automatically.

The operating system combines the UID and HID to identify the dongle and install its driver. When a dongle is first plugged into the computer, you will need to go through the driver installation process. When you reset the UID, the computer will see it as a new device and you will have to go through the process again. Once the UID is set the computer will record the new

UID/HID combination and you will not have to reinstall the driver. Keep in mind that since each dongle has a unique HID, and the operating system will check the combination of UID and HID, every new dongle inserted into the computer will require installation of its driver. However, once the driver is installed for a specific UID/ HID combination, the dongle may be removed and reinserted to the computer without reinstalling the driver.

Please note that the driver installation process is triggered by the dongle being inserted into the computer. When you change the UID, please remove and reinsert the dongle to install the driver for the new UID/HID combination.

ROCKEY2 is initially preformatted with UID=0 and the HID shielded so that HID=0. The computer operating system will see these "out of the box" dongles as device "00". You should reset the UID so that the operating system can see a unique UID/HID combination. The UID needs to be reset for multiple dongles to work together on the same computer. The UID should certainly be reset before the dongle is re-shipped and the UID generation call should not be included in the package sent to the end users.

Chapter 3 ROCKEY2 API

1. RY2_Find: Find ROCKEY2 dongles attached to the computer

EXTERN_C int WINAPI RY2_Find();

Return Value	<0	Error code
	=0	No ROCKEY2 dongle is attached
	>0	The number of attached ROCKEY2 dongle(s)

2. RY2_Open: Open specified ROCKEY2 dongle

EXTERN_C int WINAPI RY2_Open(int mode, DWORD uid, DWORD* hid);

Input	mode	This parameter indicates the way to open the dongle mode = 0, open the first found ROCKEY2 dongle mode > 0, open the dongle according to the UID. The mode value is the dongle number, for example: uid=12345678, mode=2, this means it will open the second dongle with UID 12345678 mode = -1, open the dongle according to the HID, and *hid can not be 0 We defined two MACROs: AUTO_MODE=0 and HID_MODE=-1
	uid(User ID)	You need to specify the dongle UID and this UID is generated with RY2_GenUID
	hid(Hardware ID)	Open dongle with HID of *hid The dongle HID will be returned to *hid regardless of how the dongle was opened.
Return	>=0	Success. The opened dongle handle is returned.
	< 0	Error code. Please refer to the Chapter 4 Error Codes

3. RY2_Close: Close specified ROCKEY2 dongle

EXTERN_C int WINAPI RY2_Close(int handle);

Input	ROCKEY2 dongle handle. It is the handle returned from RY2_Open
Return	Error code. Please refer to the Chapter 4 Error Codes

4. RY2_GenUID: Generate User ID(UID)

EXTERN_C int WINAPI RY2_GenUID(int handle, DWORD* uid, char* seed, int isProtect)

Input	Handle	Dongle handle. It is the handle returned from RY2_Open
	Uid	Output parameter. The generated UID
	Seed	Seed to generate UID. It is a character string with the maximum length of 64 bytes
	isProtect	Sets write protection. 0=dongle not write protected, 1=dongle write protected. We defines two MACROs: #define ROCKEY2_DISABLE_WRITE_PROTECT 0 #define ROCKEY2_ENSABLE_WRITE_PROTECT 1
Return	Error code. Please refer to Chapter 4 Error Codes	

5. RY2_Read: Read dongle content

EXTERN_C int WINAPI RY2_Read(int handle, int block_index, char* buffer512);

Input	handle	Dongle handle. It is the handle returned from RY2_Open
	block_index	Block index. Specify the block to read. The value range is 0-4
	buffer512	Read buffer. The buffer must be at least 512 bytes to accommodate the 512 byte block size.
Return	Error code. Please refer to Chapter 4 Error Codes	

6. RY2_Write: Write to ROCKEY2 dongle

EXTERN_C int WINAPI RY2_Write(int handle, int block_index, char* buffer512);

Input	handle	Dongle handle. It is the handle returned from RY2_Open
	block_index	Block index. Specify the block to write. The value range is 0-4
	buffer512	Write buffer. The buffer must be at least 512 bytes to accommodate the 512 byte block size.
Return	Error code. Please refer to Chapter 4 Error Codes	

Chapter 4 Error Codes

MACRO	Value	Description
RY2ERR_SUCCESS	0	Success
RY2ERR_NO_SUCH_DEVICE	0xA0100001	Specified dongle is not found (parameter error)
RY2ERR_NOT_OPENED_DEVICE	0xA0100002	Need to call RY2_Open first to open the dongle, then call this function (operation error)
RY2ERR_WRONG_UID	0xA0100003	Wrong UID(parameter error)
RY2ERR_WRONG_INDEX	0xA0100004	Block index error (parameter error)
RY2ERR_TOO_LONG_SEED	0xA0100005	Seed character string is longer than 64 bytes when calling GenUID (parameter error)
RY2ERR_WRITE_PROTECT	0xA0100006	Tried to write to write-protected dongle(operation error)

RY2ERR_OPEN_DEVICE	0xA0100007	Open device error (Windows error)
RY2ERR_READ_REPORT	0xA0100008	Read record error(Windows error)
RY2ERR_WRITE_REPORT	0xA0100009	Write record error(Windows error)
RY2ERR_SETUP_DI_GET_ DEVICE_INTERFACE_DETAIL	0xA010000A	Internal error (Windows error)
RY2ERR_GET_ATTRIBUTES	0xA010000B	Internal error (Windows error)
RY2ERR_GET_PREPARSED_DATA	0xA010000C	Internal error (Windows error)
RY2ERR_GETCAPS	0xA010000D	Internal error (Windows error)
RY2ERR_FREE_PREPARSED_DATA	0xA010000E	Internal error (Windows error)
RY2ERR_FLUSH_QUEUE	0xA010000F	Internal error (Windows error)
RY2ERR_SETUP_DI_CLASS_DEVS	0xA0100010	Internal error (Windows error)
RY2ERR_GET_SERIAL	0xA0100011	Internal error (Windows error)
RY2ERR_GET_PRODUCT_STRING	0xA0100012	Internal error (Windows error)
RY2ERR_TOO_LONG_DEVICE_DETAIL	0xA0100013	Internal error
RY2ERR_WRONG_REPORT_LENGTH	0xA0100020	Unknown device(hardware error)
RY2ERR_VERIFY	0xA0100021	Verification error(hardware error)
RY2ERR_UNKNOWN_ERROR	0xA010FFFF	Unknown error(hardware error)