FEITIAN

iR301-U Smart Card Reader Android

Developer Guide

Revision History:

Date	Revision	Description
April. 2013	V1.0	Release of the first version
Nov, 2013	V1.1	Update DUKPT API in document

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 hardware 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 hardware 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.

Contents

Chapter 1. Overview **1**

Chapter 2. New features **2**

Chapter 3. Definitions **3**

Error codes 3

Chapter 4. API Reference **4**

3.1 ft_reader 4
3.2 open 4
3.3 PowerOn 5
3.4 PowerOff 5
3.5 getAtr 5
3.6 getCardStatus 6
3.7 transApdu 7
3.8 getManufacturerName 7
3.9 getReaderName 8
3.10 close 8
3.11 FtInitDukpt（private interface, need OEM firmware） 9
3.12 SetEncrypt（private interface, need OEM firmware） 9
3.13 GetKeySn（private interface, need OEM firmware） 10

Chapter 1. Overview

This chapter describes how to develop iR301U reader applications, including the development interfaces supported by the product (iR301U) and how to develop applications based on these interfaces.

FEITIAN iR301-U is specially engineered to accommodate a range of smart card applications. Developers use it as a platform to generate and deploy related products and services. Moreover, FEITIAN iR301-U is a terminal unit which is seamlessly integrated to all major systems of operation. Additional features such as the built-in inclusive support for different smart card interfaces has facilitated the wide scale and cross industry adoption of iR301-U.

iR301-U suits customers where security concerns are the most salient and satisfies the demand for a flexible solution for ID authentication, e-commerce, e-payment, information security and access control.

iR301-U and the rest of FEITIAN's line of smart card readers offer each customer a complete solution for all manner of utilizations.

Chapter 2. New features

The new reader has been published, included key management and data space.

New features:

1. More security

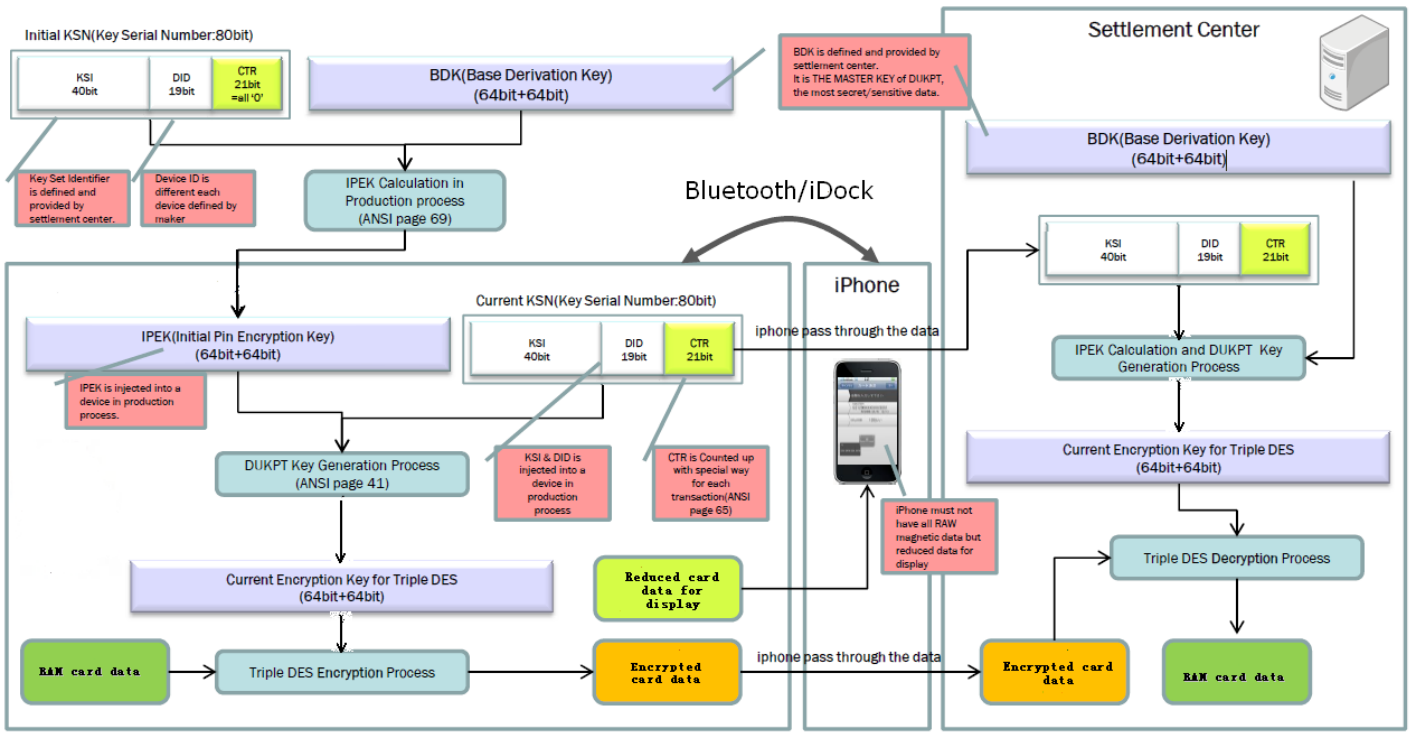
   DUKPT (Derived Unique Key Per Transaction) is a key management scheme in which for every transaction, a unique key is used which is derived from a fixed key. Therefore, if a derived key is compromised, future and past transaction data are still protected since the next or prior keys cannot be determined easily. DUKPT is specified in ANSI X9.24 part 1.

   The security standard defines card data of chip card can't be in iOS. Card reader device must be encrypting all transfer information before sending to iOS.

2. Supported 256bytes data space for customer, customer can through API to write/read data from reader.

We through below picture to give customer a clear concept of DUKPT:

http://en.wikipedia.org/wiki/DUKPT



Chapter 3. Definitions

Error codes

The following is a list of commonly used errors. Since different cards produce different errors they must map over to these error messages.

ERROR CODE:
- RETURN_SUCCESS = 0;
- RETURN_ERROR = -1;
- ERROR_RECEIVE_LRC = 240;
- PROTO_NOT_SUPPORT = 1;
- IFD_COMMUNICATION_ERROR = 612;
- IFD_NOT_SUPPORTED = 614;
- TRANS_RETURN_ERROR = 61441;
- BUFFER_NOT_ENOUGH = 61442;
- SENDERROR = 57345;

CARD STATUS:
- CARD_PRESENT = 0;
- CARD_UNKNOW = 1;
- CARD_ABSENT = 2;

READER STATUS:
- READER_PRESENT = 0;
- READER_ABSENT = 2;

CARD PROTOCOL:
- CARD_PROTOCOL_T0 = 0;
- CARD_PROTOCOL_T1 = 1;

Chapter 4. API Reference

3.1 ft_reader

Synopsis:
public ft_reader(UsbManager mUsbManager, UsbDevice mDevice)
Parameters:
N/A
Description:
Constructor Detail.

Example:
Please follow sample code.

Returns:
RETURN_SUCCESS = 0;
RETURN_ERROR = -1;
ERROR_RECEIVE_LRC = 240;

3.2 open

Synopsis:
public int open()
Parameters:
N/A
Description:
Open device.

Example:
Please follow sample code.

Returns:
RETURN_SUCCESS = 0;
RETURN_ERROR = -1;

3.3 PowerOn

Synopsis:
public int PowerOn()
Parameters:
N/A
Description:
This function use to provide power to card and get card ATR, if no card insert to reader, will return failed.

Example:
Please follow sample code.

Returns:
RETURN_SUCCESS = 0;
PROTO_NOT_SUPPORT = 1;
IFD_COMMUNICATION_ERROR = 612;
TRANS_RETURN_ERROR = 61441;

3.4 PowerOff

Synopsis:
public int PowerOff()

Parameters:
N/A
Description:
This function use to power off card.

Example:
Please follow sample code.

Returns:
RETURN_SUCCESS = 0;
PROTO_NOT_SUPPORT = 1;
IFD_COMMUNICATION_ERROR = 612;
TRANS_RETURN_ERROR = 61441;

3.5 getAtr

Synopsis:

public byte[] getAtr()

Parameters:
N/A

Description:
This function returns card ATR number.

Example:
Please follow sample code.

Returns:
RETURN_SUCCESS = 0;
PROTO_NOT_SUPPORT = 1;
IFD_COMMUNICATION_ERROR = 612;
TRANS_RETURN_ERROR = 61441;

3.6 getCardStatus

Synopsis:
public int getCardStatus()

Parameters:
N/A

Description:
Get card status:
CARD_PRESENT = 0;
CARD_UNKNOW = 1;
CARD_ABSENT = 2;

Example:
Please follow sample code.

Returns:
CARD_PRESENT = 0;
CARD_UNKNOW = 1;
CARD_ABSENT = 2;

3.7 transApdu

Synopsis:

public int transApdu(int tx_length,
byte[] tx_buffer,
int[] rx_length,
byte[] rx_buffer)

Parameters:

tx_length	IN	input data's length
tx_buffer	IN	input data
rx_length	OUT	return data's length
rx_buffer	OUT	return data from card

Description:

This function sends an APDU to the smart card contained in the reader. The card responds from the APDU and stores this response in rx_buffer and it's length in rx_length.

Example:

Please follow sample code.

Returns:

RETURN_SUCCESS = 0;
ERROR_RECEIVE_LRC = 240;
PROTO_NOT_SUPPORT = 1;
IFD_COMMUNICATION_ERROR = 612;
IFD_NOT_SUPPORTED = 614;
TRANS_RETURN_ERROR = 61441;
BUFFER_NOT_ENOUGH = 61442;

3.8 getManufacturerName

Synopsis:

public java.lang.String getManufacturerName()

Parameters:

N/A

Description:

Return manufacturer name.

Example:

Please follow sample code.

Returns:

RETURN_SUCCESS = 0;

RETURN_ERROR = -1;

3.9 getReaderName

Synopsis:

public java.lang.String getReaderName()

Parameters:

N/A

Description:

Return reader name.

Example:

Please follow sample code.

Returns:

RETURN_SUCCESS = 0;

RETURN_ERROR = -1;

3.10 close

Synopsis:

public int close()

Parameters:

N/A

Description:

Close device.

Example:

Please follow sample code.

Returns:

RETURN_SUCCESS = 0;

RETURN_ERROR = -1;

3.11 FtInitDukpt（private interface, need OEM firmware）

Synopsis:

public int FtInitDukpt (byte[] encBuf,int nLen)

Parameters:

encBuf IN IPEK(IPEK Calculation in Production process (ANSI page 69)) buffer (the new IPEK and KSN must be used exist BDK to encryption)

nLen IN Size of encBuf

Description:

The API used to inject IPEK and KSN to reader.

The IPEK is based on computer between KSN (Key Serial Number) and BDK(Base Derivation Key). More information, please follow below documents.

Section DUKPT

http://download.ftsafe.com/files/reader/ANSIX9.24PART1-2004.pdf

Example:

Please follow sample code.

Returns:

RETURN_SUCCESS = 0;

RETURN_ERROR = -1;

3.12 SetEncrypt（private interface, need OEM firmware）

Synopsis:

SetEncrypt(boolean bEncrypt,boolean bEncFunc,boolean bEncType)

Parameters:

bEncrypt	IN		BOOL value
	1	-	ENCYRPTION
	0	-	NO - ENCRYPTION
bEncFunc	IN		BOOL value(True means)
	1	-	Use AES algorithm to encryption
	0	-	Use 3DES algorithm to encryption
bEncType	IN		BOOL value(True means encryption)
	1	-	Single directional encryption
	0	-	Bidirectional encryption

Description:

The API used to switch encryption mode and do encryption use DUKPT, it supports single and bidirectional to encryption return data or APDUs.

If set bEncrypt is true, the commutation data will be encryption between application and reader.

That's mean the application need encrypt APDU command and send to reader, reader will decryption APDU command send to card, then get card data to encryption return to application.
http://download.ftsafe.com/files/reader/ANSIX9.24PART1-2004.pdf

Example:
Please follow sample code.
Returns:
RETURN_SUCCESS = 0;
RETURN_ERROR = -1;

3.13 GetKeySn（private interface, need OEM firmware）

Synopsis:
public int GetKeySn (byte[]buffer,int[]bufferLen)
Parameters:

bufferLen	OUT	KSN's length
buffer	OUT	The KSN that get from reader

Description:
The API used to get KSN number from reader.
The KSN use to computer IPEK with BDK, once the reader return encryption data, the application need use the current KSN in reader and BDK to generate IPEK and decryption data.
More information: http://download.ftsafe.com/files/reader/ANSIX9.24PART1-2004.pdf

Example:
Please follow sample code.
Returns:
RETURN_SUCCESS = 0;
RETURN_ERROR = -1;