User Manual

Blockchain Security 2Go Starter Kit

www.infineon.com

User Manual

Blockchain Security 2Go Starter Kit

About this document

Intended audience

The target readers of this document are Blockchain developers that want to develop new applications based on hardware-based security. When reading this document, you should have

- the Blockchain Security 2Go starter kit, and
- a basic technical understanding of Blockchain technology.

Table of contents

1 Introduction

Infineon's Blockchain Security 2Go starter kit provides user credential protection with a security controller offering security on a high level. The starter kit is very generic and supports many different kinds of Blockchain technologies. If you are applying Blockchain technology in your system, the Blockchain Security 2Go starter kit allows you to seamlessly integrate hardware-based security. It provides a lean feature set as well as open source application examples, which enable new ideas to flourish and to generate a secured physical link from the digital to the real world.

The Infineon Blockchain Security 2Go starter kit provides an evaluation environment and includes:

- 5 credit card sized ID1 cards based on ISO/IEC 7810 [1] having a contactless interface and a Class 1 communication antenna based on based on ISO/IEC 14443 [2].
- An Infineon security controller.
- On-card software that supports commands for key-management, signature creation and PIN authentication
- Open-source software that exemplifies how to integrate the features of the Blockchain Security 2Go cards in a real-world Blockchain system (e.g. sending cryptocurrencies or integrating the cards in a smart contract for eVoting). The software is open source and is hosted on GitHub [3].

The main features that the Blockchain Security 2Go cards offer are

- creation and storage of up to 255 private/public key pairs for Blockchain applications,
- loading and storing a key that is provided by the user in an encrypted form,
- signature generation for signing Blockchain transactions and
- user authentication with PIN.

Figure 1 The Blockchain Security 2Go Starter Kit

To learn more about Blockchain in general we recommend the open source books from Andreas M. Antonopoulos

- "Mastering Bitcoin 2nd Edition - Programming the Open Blockchain", A. M. Antonopoulos [4] and
- "Mastering Ethereum", A. M. Antonopoulos, Gavin Wood [5].

1.1 Basic Command Overview

To get a first basic overview of the basic commands that are supported by the Blockchain Security 2Go starter kit see Table 1. Chapter 4 provides more details about what the commands do and how they are structured.

Table 1 Overview of the Basic Commands

Command	Short Description
GENERATE KEY	Generates and stores a new public/private keypair.
ENCRYPTED KEYIMPORT	Creates and stores a new public/private key pair by deriving the private key from a given seed.
GET KEY INFO	Provides information about a specific keypair such as the public key.
GENERATE SIGNATURE	Generates a signature of a given hash (32 bytes).

2 How to Integrate the Starter Kit in Blockchain Applications

In a Blockchain economy, there are various participants such as miner, nodes and the end users as shown in Figure 2. Basically, a Blockchain is a decentralized digital ledger that manages a continuously growing list of data points (chain of blocks). Every block in the chain is cryptographically linked to the previous block. Consequently, to change one block and remain validity, an attacker would have to change the entire chain. The ledger records all transactions that have been sent to or from different accounts. This transaction history allows users to determine the current asset value that belong to an account.

All transactions are protected by a digital signature. This makes it extremely difficult to change or alter them without being detected. To create such a digital signature, a secret private key that corresponds to the public key (address) of an account is needed. When knowing the private key of a user, an attacker can do a lot of damage as it allows the creation of seemingly valid transactions. Typically, there is no third party and no possibility to alter the history of a Blockchain, there is no way to revoke such a transaction. Therefore, keys (i.e. Blockchain credentials) require strong protection level in terms of security. This can be achieved with the Blockchain Security 2Go starter kit.

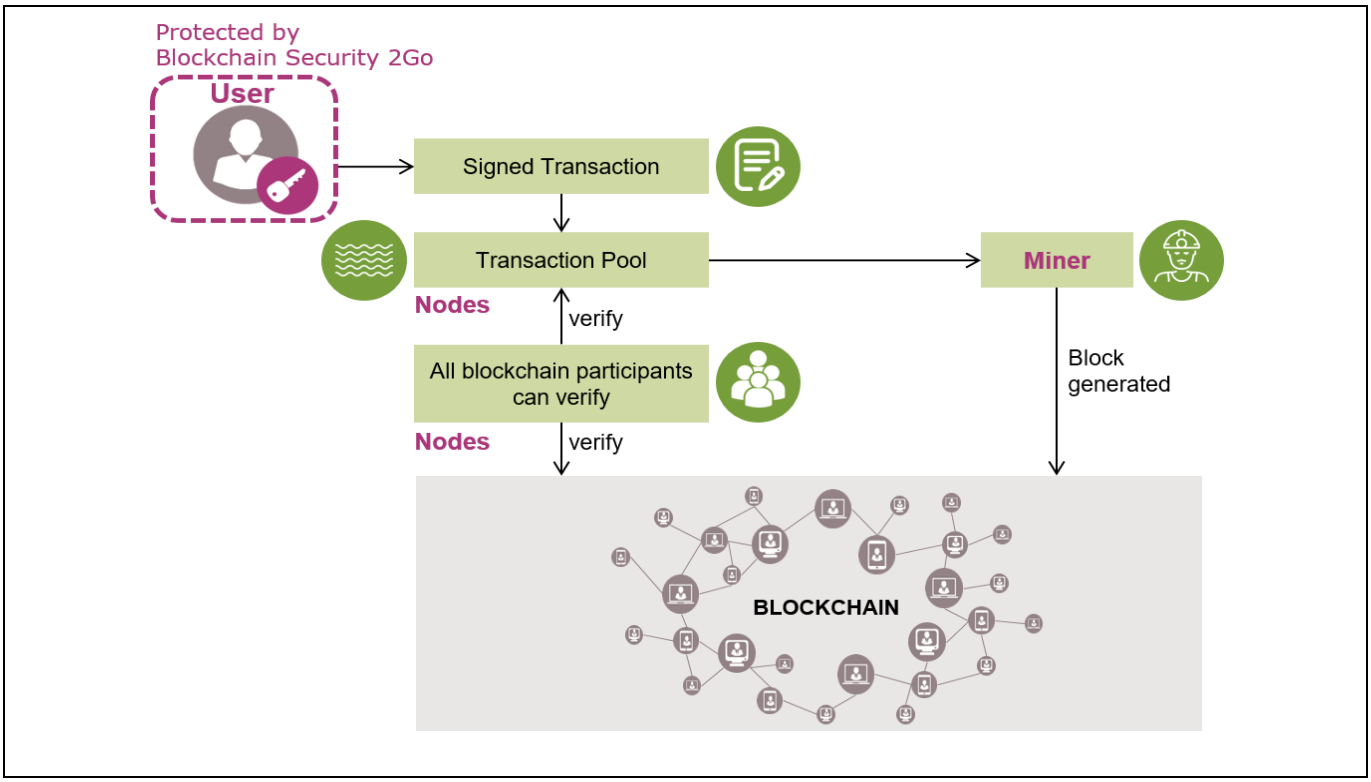

Figure 2 The Blockchain Security 2Go starter kit offers protection for the user credentials (keys).

To link the Blockchain Security 2Go smart cards to a Blockchain, you need an interface device that handles the communication with the Blockchain (see Figure 3). This could either be

- an NFC-enabled smartphone, or
- a host device (e.g. PC, RasperryPi) connected to a contactless reader (e.g. via a PC/SC interface).

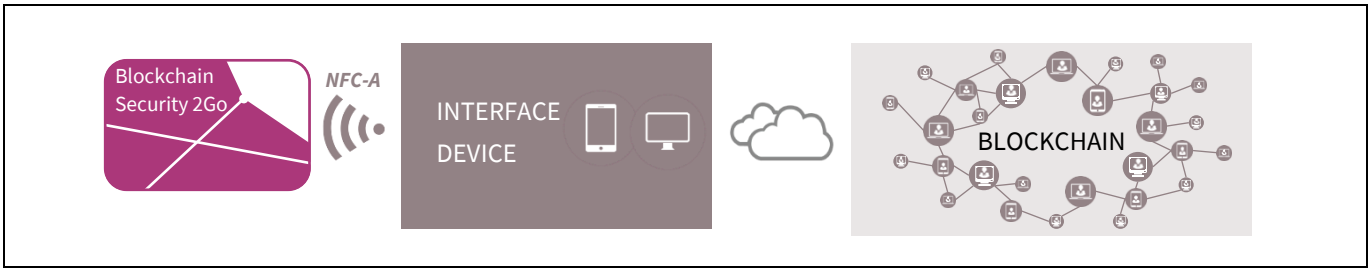

Figure 3 An interface device having the capability to communicate via NFC to the card and via a network (e.g. internet) to a Blockchain creates the link between the Blockchain Security 2Go cards and the Blockchain network.

2.1 Supported Blockchains

The vast majority of currently existing Blockchains use Elliptic-Curve Cryptography (ECC) as an asymmetric cryptography method. Typically, the elliptic curve *secp256k1* is used [6]. The Blockchain Security 2Go starter kit supports all Blockchains based on ECC using the *secp256k1* curve, regardless of other underlying technologies such as the network, the form of the Blockchain (public, private), or the application. Some selected examples of supported existing Blockchains are

- Bitcoin,
- Ethereum and all ERC-20 tokens,
- and many more.

2.2 User Credentials Creation

To create new user credentials the interface device has to first trigger a new key generation on the card which results in the card securely creating and storing a new public-private key pair. The public key is used to derive the address of the new account at which point transactions to the newly generated address and the account is ready to participate in the Blockchain

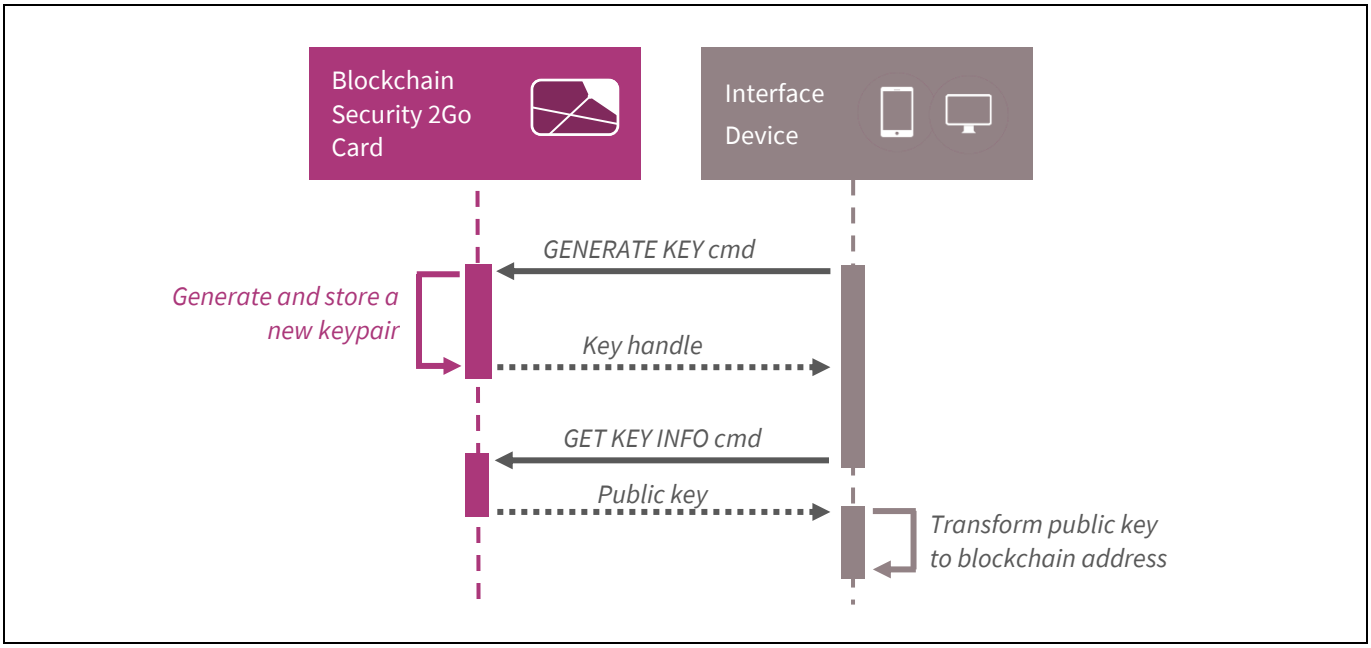

Figure 4 **Generation of a new Blockchain address with the Blockchain Security 2Go starter kit. Note, instead of the GENERATE KEY command also the ENCRYPTED KEYIMPORT command could be used to generate a new key.**

2.3 Transaction Signing

Typically, in Blockchain systems transactions are used to send assets (e.g. cryptocurrency) from one account to another.

To demonstrate the authenticity of the sender, the transaction is signed with the senders' private key. Other participants (i.e. nodes) on the Blockchain use the public key of the sender to verify that the transaction is authentic before adding the transaction to a new block in the Blockchain.

Usually a transaction includes information such as the receiver's public key, the amount of assets that should be transferred or arbitrary data for a smart contract. Before the transaction is signed, the transaction is hashed (e.g. SHA-256 [7]). Then, the signature of this hashed data is calculated on the card with the senders' private key (see Figure 5). To calculate a signature with a Blockchain Security 2Go card, it has to be hashed off-card. The Blockchain Security 2Go starter kit supports all hashes that lead to 32 byte output data.

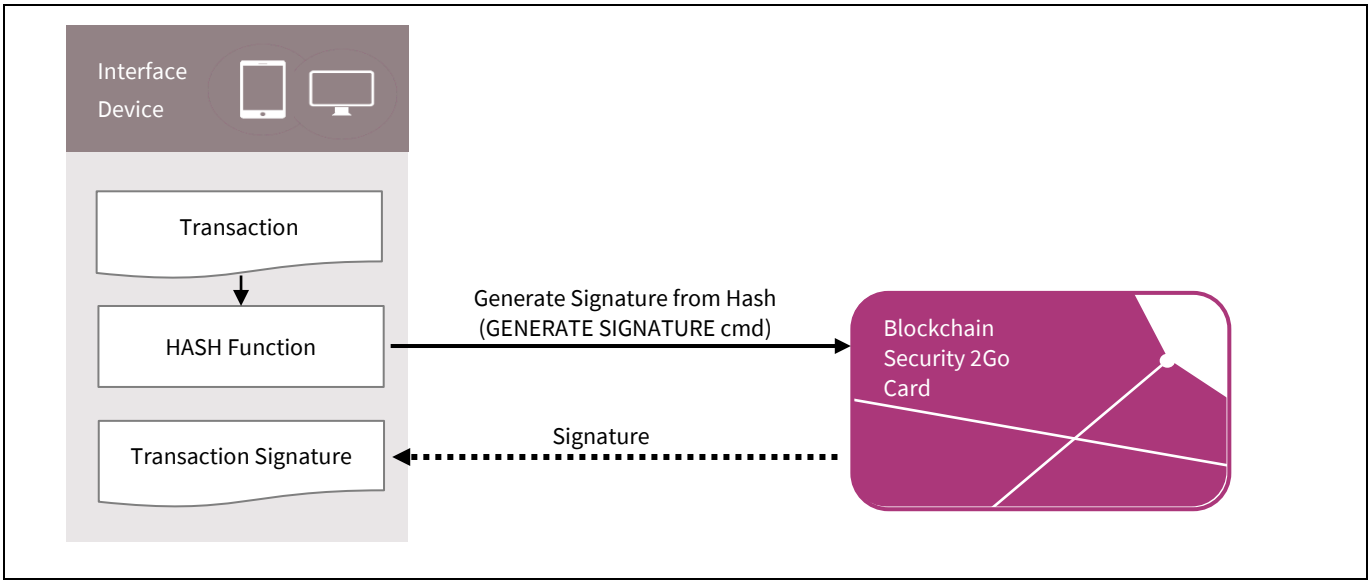

Figure 5 **To generate a signature of a transaction, first the transaction message is hashed, then the Blockchain Security 2Go card calculates a signature of this hashed message.**

3 How to Use the Starter Kit

3.1 Key Management

In Blockchain systems, knowing a secret private key is directly associated with the control rights for an account. Consequently, it is important to protect the private key. The Blockchain Security 2Go cards feature hardware-based protection mechanisms to generate and store private keys in a secured way.

Note: Always keep the private keys highly protected with hardware-based security, there is no export feature for the private keys.

One Blockchain Security 2Go card can generate and store 255 private-public key pairs. Additionally, it is possible to import a keypair that is derived from a password (seed) that is provided by the user. This is achieved with the encrypted keyimport feature. To get the public-key of a keypair use the GET KEY INFO command.

The Blockchain Security 2Go cards support on-card key generation providing highly secured private keys. This is achieved with a high entropy hardware-based random number generator. To do this use the GENERATE KEY command.

The encrypted keyimport feature allows that on different Blockchain Security 2Go cards the same private key is generated. The user can provide a password (seed). From that given seed a private key is derived with the standardized key derivation function as defined in NIST SP 800-108 [8] using CMAC-AES256 as defined in NIST SP 800-35B [9].This allows that multiple cards can store the same private key. However, the private key itself is derived and stored on-card and is not known to the user.

To identify the keypairs, each pair is associated with a keyhandle. An imported key always belongs to the keyhandle 0.

Table 2 Overview of the keyhandles for identifying keypairs

Keyhandle	Number of supported keypairs	Command for generation	Description
1-255	255	GENERATE KEY	On-card generated keypairs
0	1	ENCRYPTED KEYIMPORT	Imported keypair

3.2 Creation of Backups

To create a backup of a keypair use the ENCRYPTED KEYIMPORT command on different cards and provide the same seed. This will lead to the same private key that is stored on the different cards having the keyhandle 0 (see Figure 6).

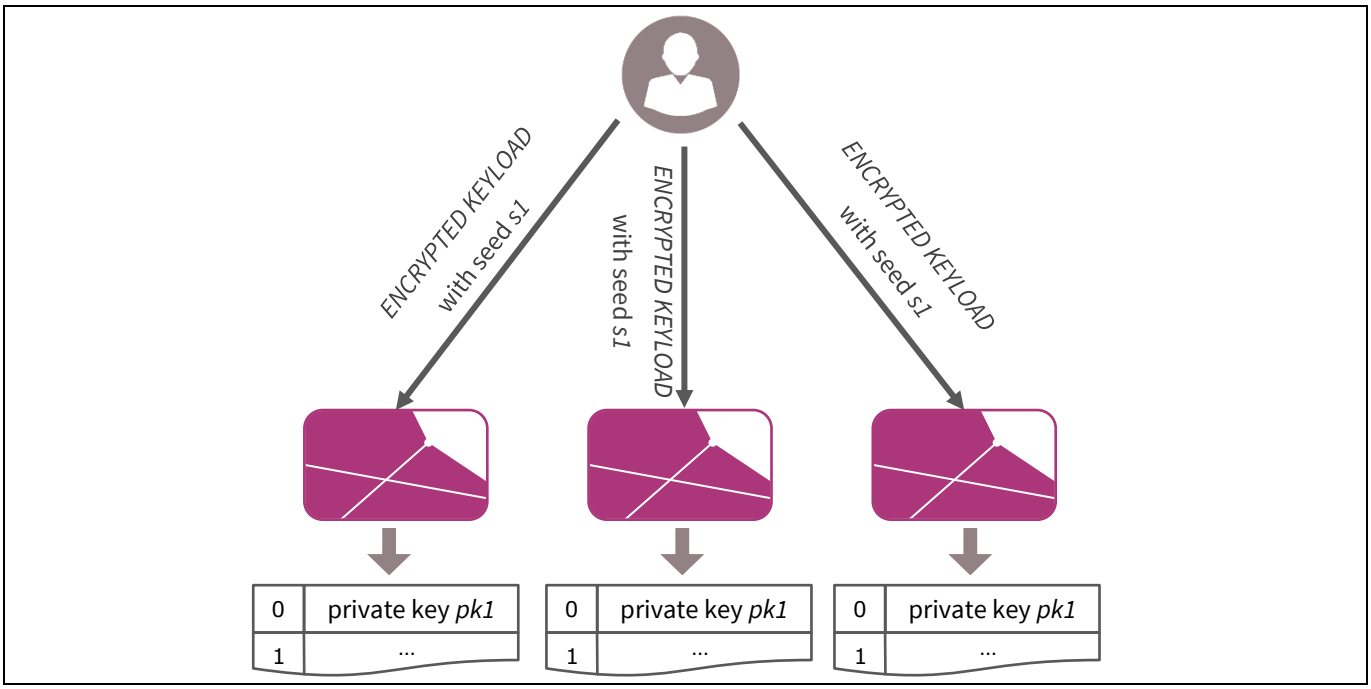

Figure 6 **Using the ENCRYPTED KEYLOAD command with the same seed on different cards will lead to the same keypair at each card. This allows to make backups for the imported key.**

Attention: ***It is not possible to directly backup on-card generated keys that are created with the GENERATE KEY command. These keys cannot be exported or cloned. It will be necessary to use backup mechanisms at a higher level, such as generating multi-signature accounts.***

3.3 PIN Authentication

The authentication with PIN makes malicious misuse of a Blockchain Security 2Go card harder. Only persons that know the secret PIN value are allowed to use PIN protected functions. This offers protection of the user assets for example in case the card is lost or stolen.

The PIN authentication is optional. The user can decide whether to use it or not. If no PIN is configured, all commands are allowed without a preceded PIN authentication. Once the user configures the card with a new PIN using the SET PIN command, the PIN has to be provided before executing the commands

- GENERATE SIGNATURE, and
- ENCRYPTED KEYLOAD.

3.3.1 PIN commands

Below, Table 3 gives an overview of the commands that are used to manage the authentication with PIN. Chapter 4.3.3 provides more details about the PIN commands.

Table 3 Overview of the PIN Commands

Command	Short Description
SET PIN	Initializes PIN authentication by setting a PIN value and returns a PUK value.
CHANGE PIN	Changes the current PIN value and creates a new PUK value.
VERIFY PIN	Activates a new PIN session, so that PIN protected commands are allowed.
UNLOCK PIN	Deactivates the PIN authentication when providing the valid PUK value.

3.3.2 PIN Authentication Procedure

Figure 7 illustrates the overall procedure of the PIN authentication. Initially, all commands are allowed until a PIN is set via the SET PIN command. Once the PIN is set, the user has to initialize a valid PIN session with the VERIFY PIN command. If the card is in such an active PIN session, all commands are allowed until the session is invalidated. A PIN session automatically ends whenever the card is removed. Additionally, the SELECT APP command closes an active PIN session. This allows applications to invalidate a PIN session in use-cases where a card remains on a reader for a long time period.

To prevent attacks where an attacker tries to guess the PIN, the number of consecutive incorrect PIN entries is limited as outlined in Table 4. Each time a wrong PIN is entered (i.e. by using the VERIFY PIN or CHANGE PIN command) a PIN retry counter is decreased. Once the counter indicates that the maximal number of wrong retries is reached, the PIN is locked. Then, the VERIFY PIN and CHANGE PIN commands are not allowed anymore. Consequently, it is not possible to activate a new PIN session and use PIN protected commands.

To unlock the card use the UNLOCK PIN command and give the valid PUK value. This deactivates the PIN feature and as long as no new PIN is set with the SET PIN command. Each time a new PIN is configured (i.e. with the SET PIN or CHANGE PIN commands), a new PUK value is generated and returned. This PUK code has to be remembered for the case that the PIN has been entered wrong too many times. If the PUK has been entered wrong too many times, the card is irreversibly locked.

Attention: ***It is important to guide the user to carefully backup the PUK value when setting a PIN. When losing the PUK value it is not anymore possible to use a locked card.***

Table 4 Maximal number of wrong PIN and PUK entries

	Allowed retries
PIN	3
PUK	6

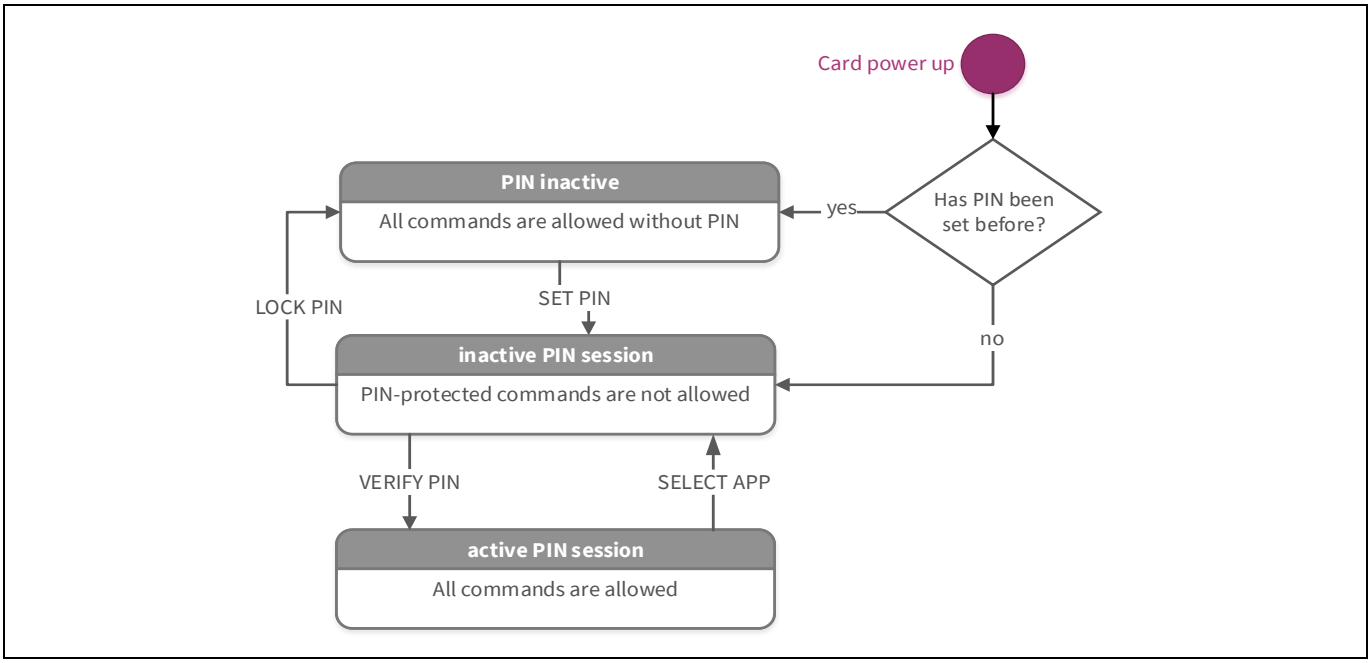

Figure 7 Overview of the PIN States

3.4 Expiring Operations

To enhance the security of the Blockchain Security 2Go card, specific operations have a usage limitation. This means that after calling these operations for a certain amount of times, the operations are blocked.

The number of signatures that can be generated with a keypair is limited as outlined in Table 5. The global signature counter and the key-specific signature counters indicate the remaining number of allowed signature operations. Every signature key usage decrements these counters. If a key-specific counter has expired it is not possible to use the effected private key anymore. If the global signature counter expires, no private key can be used from that point forward. The GENERATE SIGNATURE command as well as the GET KEY INFO command return these counters in their responses. This allows users to keep track of the current usage counters and warn the end user if as the counter approaches zero.

Attention: Once one of the signature counters reaches zero, it is not possible to use the affected keys from that point forward. Thus it is important to provide an appropriate mechanism to keep the assets that are associated with the keys and for example, to transfer the assets to another account.

Table 5 Expiring signature generations

Expiring operation	Limitation counter	Maximal value
Signatures that can be generated with one card	Global signature counter	1 000 000
Signatures with each on-card generated key (keyhandles between 1 and 255)	Signature counter	100 000
Signatures with the imported key (key handle 0)		10 000

Additionally, the number of encrypted keyimports is limited as shown in Table 6.

Table 6 Expiring ENCRYPTED KEYIMPORT calls

Expiring operation	Maximal value
Number of ENCRYPTED KEYIMPORT command calls	255

3.5 Contactless Communication Interface

The Blockchain Security 2Go starter kit uses the contactless interface and near-field communication (NFC- Type A) to interact with a reader device. NFC is characterized as a short-range communication technology that allows the contactless communication between a NFC reader and a passive (no battery) device such as a Blockchain Security 2go card. Beside standard NFC reader devices which are connected typically via USB to a PC, nowadays the majority of smartphones include NFC functionality.

The Blockchain Security 2Go starter kit contactless communication is based on ISO/IEC 14443 [2]. The main components of the cards are:

- The plastic carrier with the ID-1 format specified in ISO/IEC 7810 [1]. ID-1 is commonly used for standard payment cards or a driving license and has the dimensions of 85.60 × 53.98 mm and rounded corners with a radius of 2.88–3.48 mm.
- The Infineon security controller.
- The Class 1 communication antenna based on ISO/IEC 14443-1 [10] which is connected to the chip package. The ISO Class 1 shape classification is a square with the outline dimensions of 64-81 mm and 34-49 mm

Note: *The Blockchain Security 2Go card dimension and antenna arrangement represents just one of the many form factors that are supported by Infineons' security controller. Certainly other form factors which are using the contactless interfaces would be feasible too. Examples for innovative contactless chip integrations are provided by the company NFCRing [11] and by Infineon's SECORA™ Pay W portfolio [12].*

To establish a robust contactless interface channel position the card antenna above the reader device. As illustrated in Figure 8, the effective area of the reader antenna as well as the area of the card antenna should overlap. Thus the magnetic field, generated by the reader antenna, is able to flow through the card antenna.

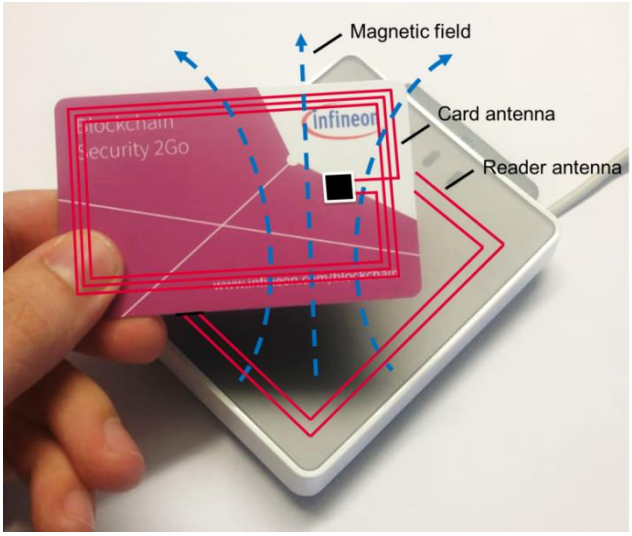

Figure 8 **Positioning of the Blockchain Security 2Go card on a reader. The class1 antenna of the card should be positioned in a way that the electromagnetic field, generated by the reader antenna flows through the inner area of the card antenna.**

In contrast to a standard NFC USB reader, the reader antenna of mobile devices is not always located around the center. If you have troubles when communicating with the Blockchain Security 2Go cards, investigate the positioning of the card in combination with the used NFC phone. Figure 9 shows the backside of a Google Pixel smartphone. In this case the NFC antenna is located in the upper area of the smartphone. For more details check out the hardware diagram of the Pixel smartphone [13].

Figure 9 Ideal inductively coupled Blockchain Security 2Go Card with a Google Pixel smartphone. The glossy area of the phone indicates the NFC detection area.

4 API

4.1 APDUs

Application Protocol Data Units (APDUs) represent the standard communication messaging format between a smart card and an application device and is defined in part 4 of the ISO 7816 standard [14]. There are two types of APDUS:

- Command APDUs (to send commands to a smart card) (see Table 7), and
- Response APDUs (to receive answers from a smart card) (see Table 8).

The communication is always initialized from the application device.

Table 7 Command APDU

	Identifier	Name	Length (in bytes)	Meaning
header	CLA	Class	1	Class of the instruction
	INS	Instruction	1	Instruction
	P1	Parameter 1	1	Parameter for the command
	P2	Parameter 2	1	Parameter for the command
body	Lc	Length command	0 to 1	Length of the command data
	Data	Data	Lc	Command data
	Le	Length expected	0 to 1	Length of the expected answer

Table 8 Response APDU

Identifier	Length (in bytes)	Meaning
Data	N_r (at most N_e)	Response data field
SW1	1	Status word
SW2	1	Status word

The Lc field gives the length of the data in the command. If there is no data, the length byte is absent.

The Le parameter in the command APDU denotes the number of expected bytes in the response data field as follows:

- If Le is absent, then Ne is zero.
- A short Le field consists of one byte with any value.
    - From 0x01 to 0xFF, the byte encodes Ne from one to 255.
    - If the byte is set to 0x00, then Ne is 256. Here, we use 0x00 to indicate that it is not required to explicitly limit the length of the expected answer.
- An extended Le field consists of either three or two bytes (for more details see [15]). The Blockchain Security2 go cards do not require to use extended Le fields.

4.2 Error Values

Table 9 outlines the most common errors regarding the communication and APDU formatting (for more details see ISO 7816-3 [15] and ISO 7816-4 [16]). Command-specific errors are listed in the corresponding command description below.

Table 9 Generic ISO 7816 errors

Error Code	Description
90 00_H	Success
64 XX_H	Operation failed (further information in XX)
67 00_H	Wrong length
6A 86_H	Incorrect parameters P1/P2
6D 00_H	Instruction code is not supported or invalid or application has not selected with the SELECT APP command
6E 00_H	Class not supported
6F 00_H	Unknown error

4.3 Command Details

4.3.1 Select Application

The SELECT APP command must be sent as a first command to initialize the current application session.

Attention: ***Before successfully sending the SELECT APP command, no other commands will work.***

Additionally, it provides information about the status of the card:

- The PIN activation status indicates whether a PIN has been set or not.
- The card ID is a unique identifier for each card and can be used to identify a specific card.

The Application Identifier (AID) for the Blockchain Security 2Go starter kit is

```
D2 76 00 00 04 15 02 00 01 00 00 00 01
```

Table 10 SELECT APP Command

Code	Value	Meaning
CLA	00_H	
INS	$A4_H$	SELECT APP
P1	04_H	Select by DF name
P2	00_H	No information given
Lc	$0D_H$	Length of data field
Data	AID *(13 bytes)*	AID value of the Blockchain Security 2Go starter kit
Le	12_H	Length of the expected answer

Table 11 SELECT APP Response

Data	Length (in bytes)	Meaning
PIN activation status	1	Indication whether PIN is active (1) or inactive (0)
ID	10	Card ID
VersionString	7	Information about the current version (ASCII encoded)
90 00	2	Success

Table 12 SELECT APP Error Codes

Data	Meaning
6A 82_H	Selected Application not found – wrong AID

4.3.2 Basic Commands

4.3.2.1 GENERATE KEY

Create new ECC private/public keypair.

Note: Whenever a new key is generation is triggered a new keyhandle is created. However, if there is a communication error during key generation, a keyhandle might be corrupted and cannot be used afterwards. This means that whenever a user tries to use such a corrupted keypair (i.e. with the GET KEY INFO or GENERATE SIGNATURE command) the command result in the error 0x6A88.

Table 13 GENERATE KEY Command

Code	Value	Meaning
CLA	00_H	
INS	02_H	GENERATE KEY Command
P1	00_H	RFU
P2	00_H	RFU
Le	01_H	Expected length of answer

Table 14 GENERATE KEY Response

Data	Meaning
01_H …FF_H	Key handle between 0x01 and 0xFF
90 00	Success

Table 15 GENERATE KEY Error Codes

Data	Meaning
6A 84_H	Key storage is full (not enough memory)

4.3.2.2 ENCRYPTED KEYIMPORT

Create a new key pair by deriving the private key from a given seed. The encrypted key is associated with the key handle 0.

The key derivation operates according to NIST SP800-108 using CMAC-AES256 as defined in NIST SP 800-38B is used. The generated private key based on the keyimport seed depends on a secret value that is already stored in the card.

Attention: ***Note, if a key has already been imported, again calling this command leads to overwriting the existing imported key.***

Attention: ***The number of key import calls is limited. When exceeding this number, no new key will be generated. However, the existing key that has been generated by the last keyimport can still be used.***

Table 16 ENCRYPTED KEYIMPORT Command

Code	Value	Meaning
CLA	00_H	
INS	20_H	ENCRYPTED KEYIMPORT
P1	00_H	RFU
P2	00_H	RFU
Lc	16_H	Length of seed
Data	Seed *(16 bytes)*	User seed for deriving a key pair

Table 17 ENCRYPTED KEYIMPORT Response

Data	Meaning
90 00_H	Success

Table 18 ENCRYPTED KEYIMPORT Error Codes

Data	Meaning
69 82_H	Maximal number of key import calls exceeded (Security status not satisfied)
69 85_H	Not authenticated with PIN (Condition of use not satisfied)

4.3.2.3 GET KEY INFO

Returns the public key of a given key handle. Additionally, the current signature counter for the given key and the global signature counter are returned.

Table 19 GET KEY INFO Command

Code	Value	Meaning
CLA	00_H	
INS	16_H	GET KEY INFO
P1	Key handle *(1 byte)*	Key handle between 0 and 255
P2	00_H	RFU
Le	00_H	

Table 20 GET KEY INFO Response

Data	Length (in bytes)	Meaning
Global signature counter	4	Remaining signatures of the card (unsigned, MSB first)
Signature counter	4	Remaining signatures for the given key (unsigned, MSB first)
Public-key	65	Sec1 encoded uncompressed public key [17] *04 || x-coordinate (32 bytes) || y-coordinate (32 bytes)*
90 00_H	2	Success

Table 21 GET KEY INFO Error Codes

Data	Meaning
6A 88_H	Given key handle is not available (Referenced data not found)

4.3.2.4 GENERATE SIGNATURE

Signs a given block of data using the stored private key that is associated with the given key handle.

Note: The signature will always be returned in canonical form.

Table 22 GENERATE SIGNATURE Command

Code	Value	Meaning
CLA	00_H	
INS	18_H	GENERATE SIGNATURE
P1	Key handle *(1 byte)*	Key that should be used to generate the signature
P2	00_H	RFU
Lc	20_H	Length of data
Data	Data to sign *(32 bytes)*	Data that should be signed
Le	00_H	

Table 23 GENERATE SIGNATURE Response

Data	Length (in bytes)	Meaning
Global signature counter	4	Remaining signatures of the card (unsigned, MSB first)
Signature counter	4	Remaining signatures for the given key (unsigned, MSB first)
Signature	Variable	ASN.1 DER [6] encoded signature (see below)
90 00_H		Success

Table 24 ASN.1 DER Signature Encoding Details (for more information see RFC 3279 [6])

30	lt	02	lr	r_0 r_1 … r_{lr-1}	02	ls	s_0 s_1 … s_{ls-1}
DER TAG Signature (0x30)	Total length of signature	DER TAG component (0x02)	Length or R (bytes)	R component	DER TAG component	Lenght of S (bytes)	S component

Table 25 GENERATE SIGNATURE Error Codes

Data	Meaning
69 82_H	Global or key-specific signature counter exceeded (Security status not satisfied)
69 85_H	Not authenticated with PIN (Condition of use not satisfied)
6A 88_H	Key with given key handle is not available (Referenced data not found)

4.3.3 PIN Commands

4.3.3.1 PIN and PUK Format

No special format for the PIN is required as the given binary value is used as is.

- Minimal PIN length: 4 bytes
- Maximal PIN length: 62 bytes

The PUK is an 8 bytes value.

4.3.3.2 SET PIN

Initial set up of the PIN.

Table 26 SET PIN Command

Code	Value	Meaning
CLA	00_H	
INS	40_H	SET PIN
P1	00_H	RFU
P2	00_H	RFU
Lc	L_{pin}	Length of the PIN in bytes (between 4 and 62 bytes)
Data	PIN value (*L_{pin} bytes)*	PIN value of length L_{pin} (format see below)
Le	08_H	Expected length of answer

Table 27 SET PIN Response

Data	Length (in bytes)	Meaning
PUK	8	PUK value
90 00_H	2	Success

Table 28 SET PIN Error Codes

Data	Meaning
67 00_H	PIN format is not valid (Wrong length)
69 85_H	PIN has already been set (Condition of use not satisfied)

4.3.3.3 CHANGE PIN

Change current PIN value.

Table 29 CHANGE PIN Command

Code	Value	Meaning
CLA	00_H	
INS	42_H	CHANGE PIN
P1	00_H	RFU
P2	00_H	RFU
Lc	$L_{currentPIN} + L_{newPIN} + 2$	Length of the data
Data	$L_{currPIN}$ *(1 byte)*	Length of the current PIN value in bytes
	PIN_{curr} *(between 4 and 62 bytes)*	Current PIN value
	L_{newPIN} *(1 byte)*	Length of the new PIN value in bytes
	PIN_{new} *(between 4 and 62 bytes)*	New PIN value
Le	08_H	Expected length of answer

Table 30 CHANGE PIN Response

Data	Meaning
PUK *(8 byte)*	PUK value
$90\ 00_H$	Success

Table 31 CHANGE PIN Error Codes

Data	Meaning
$63\ CX_H$	PIN is not valid, X retries remaining (Authentication failed)
$69\ 83_H$	Authentication failed, PIN blocked (Authentication method blocked)
$69\ 85_H$	PIN has not been set (Condition of use not satisfied)
$6A\ 80_H$	Format of the new PIN is not valid (min / max length), or Format of the data field not valid (i.e. lengths do not match) (Incorrect parameter in the command data field)

4.3.3.4 VERIFY PIN

Activate a new PIN session, so that commands requiring authentication are allowed.

Table 32 VERIFY PIN Command

Code	Value	Meaning
CLA	00_H	
INS	44_H	VERIFY PIN
P1	00_H	RFU
P2	00_H	RFU
Lc	L_{pin}	Length of the given PIN
Data	PIN value *(L_{pin} bytes)*	Given PIN value

Table 33 VERIFY PIN Response

Data	Meaning
90 00_H	Success

Table 34 VERIFY PIN Error Codes

Data	Meaning
63 CX_H	PIN is not valid, X retries remaining (Authentication failed)
69 83_H	Authentication failed, PIN blocked (Authentication method blocked)
69 85_H	PIN has not been set (Condition of use not satisfied)

4.3.3.5 UNLOCK PIN

Deactivates PIN authentication. This could be desired if

- user authentication has been set with SET PIN, but is not wanted any more, or
- the card blocks the authentication as a result of too many wrong PIN entries.

Table 35 UNLOCK PIN Command

Code	Value	Meaning
CLA	00_H	Base Logical channel
INS	46_H	UNLOCK PIN
P1	00_H	RFU
P2	00_H	RFU
Lc	08_H	Length of PUK value in bytes
Data	PUK value *(8 byte)*	PUK value

Table 36 UNLOCK PIN Response

Data	Meaning
90 00_H	Success

Table 37 UNLOCK PIN Error Codes

Data	Meaning
63 CX_H	PUK is not valid, X retries remaining (Authentication failed)
69 83_H	Authentication failed, PUK locked (Authentication method blocked)
69 85_H	PIN has not been set (Condition of use not satisfied)

4.4 Usage Example

Here is an example of how a sequence of commands could look like.

	Message	Meaning
→	00A404000DD2760000041502000100000000100	SELECT command
←	0002095F85000100AD00FE76312E302E30 9000	Pin activation status: “PIN inactive”, Card ID: “02095F85000100AD00FE” Version: “v1.0.0”
→	0002000000	GENERATE KEY
←	01 9000	Key handle: ”01”
→	00180100 20 A1A37394D261B648E7E257F3A6 04E328FD622910086C142A18480A027E9FF45C 00	GENERATE SIGNATURE, key handle: ”01”, data: “A1A37394D261B648E7E257F3A604E328FD6 22910086C142A18480A027E9FF45C”
←	000F423F 0001869F 304402207E191F6B8DB9069327 B4544E4E82B601BE337A45ABDB1D0114B3C2D5BEF6 8D82022010F8A9AD6B42144D85AF0C13C6F47A9D86 63A80E54743DAB1B8DF6D958CD79FD 9000	Global signature counter: “999999”, Signature counter for key 1:” 99999”, Signature: “304402207E191F6B8DB9069327B4544E4E8 2B601BE337A45ABDB1D0114B3C2D5BEF68D8 2022010F8A9AD6B42144D85AF0C13C6F47A9 D8663A80E54743DAB1B8DF6D958CD79FD”
→	00400000 04 12345678 00	SET PIN, Pin Value: “1234568”
←	56D1CDF483E9393A 9000	PUK: “56D1CDF483E9393A”
→	00180100 20 B51C9987EEB2A8B04B82F3914D 478834BBACABCD0451FC2A0BC617F17614A3A4 00	GENERATE SIGNATURE, key handle: ”01”, data: “B51C9987EEB2A8B04B82F3914D478834BBA CABCD0451FC2A0BC617F17614A3A4”
←	6985	Error (Conditions of use not satisfied) – since no PIN session is active
→	00440000 04 12345678 00	VERIFY PIN, Pin value: “12345678”
←	9000	Success
→	00180100 20 B51C9987EEB2A8B04B82F3914D 478834BBACABCD0451FC2A0BC617F17614A3A4 00	GENERATE SIGNATURE, key handle: ”01”, data: “B51C9987EEB2A8B04B82F3914D478834BBA CABCD0451FC2A0BC617F17614A3A4”
←	000F423E 0001869E 304402207FC7B1DD5027B09D52 DAEBF936430813411C8E38C11C83FD7FB4CE84BD49 967F02203F6E01487DEF0B3116B687C9A619A05EAA 4C664B772D55B5546E207797DFAF36 9000	Global signature counter: 999998, Signature counter for key 1: 99998, Signature: “304402207FC7B1DD5027B09D52 DAEBF936430813411C8E38C11C83FD7FB4CE 84BD49967F02203F6E01487DEF0B3116B687 C9A619A05EAA4C664B772D55B5546E207797 DFAF36”

5 Abbreviations

Table 38 Abbreviations

Abbreviation	Meaning
AID	Application Identifier
APDU	Application Protocol Data Unit
CLA	Class
DER	Distinguished Encoding Rules
ECC	Elliptic-Curve Cryptography
ERC	Ethereum Request for Comments
INS	Instruction
ISO	International Organization for Standardization
NIST SP	National Institute of Standards & Technology Special Publication
PC/SC	Personal Computer/Smart Card
PIN	Personal Identification Number
PUK	Personal Unlocking Key
RFU	Reserved for Future Use
SEC	Standards for Efficient Cryptography
TRNG	True Random Number Generator

6 References

[1] "ISO/IEC 7810 Identification cards - Physical characteristics," 2003. [Online]. Available: https://www.iso.org/standard/31432.html.

[2] "ISO/IEC 14443-3 Cards and security devices for personal identification - Contactless proximity objects - Part 3: Initialization and anticollision," 2016. [Online]. Available: https://www.iso.org/standard/70171.html.

[3] Infineon Technologies AG, "Blockchain Security 2Go GitHub," [Online]. Available: https://github.com/Infineon/blockchain.

[4] A. M. Antonopoulos, " Mastering Bitcoin 2nd Edition - Programming the Open Blockchain," [Online]. Available: https://github.com/bitcoinbook/bitcoinbook.

[5] G. W. Andreas M. Antonopoulos, "Mastering Ethereum," [Online]. Available: https://github.com/ethereumbook/ethereumbook.

[6] The Internet Society, "RFC 3279 Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure," 2002. [Online]. Available: https://tools.ietf.org/html/rfc3279 .

[7] NIST, "FIPS 180-4 Secure Hash Standard (SHS)," [Online]. Available: https://csrc.nist.gov/publications/detail/fips/180/4/final.

[8] "NIST SP 800-108 Recommendation for Key Derivation Using Pseudorandom Functions," 2009. [Online]. Available: https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=900147.

[9] "NIST SP 800-35B Guide to Information Technology Security Services," 2003. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-35/final.

[10] "ISO/IEC 14443-1 Cards and security devices for personal identification - Contactless proximity objects - Part1: Physical characteristics," 2018. [Online]. Available: https://www.iso.org/standard/73596.html.

[11] NFCRing, "NFCRing Website," [Online]. Available: https://nfcring.com/.

[12] Infineon Technologies AG, "SECORA™ Pay W," [Online]. Available: https://www.infineon.com/cms/de/about-infineon/press/market-news/2018/INFDSS201811-020.html .

[13] Google, "Pixel - Hardwarediagram," [Online]. Available: https://support.google.com/pixelphone/answer/7157629?hl=en.

[14] "ISO/IEC 7816 “Identification cards – Integrated circuit cards," [Online]. Available: https://www.iso.org/standard/38770.html.

[15] "ISO/IEC 7816-3 Identification cards – Integrated circuit cards Part 3," 2006. [Online]. Available: https://www.iso.org/standard/38770.html.

[16] "ISO/IEC 7816-4 Identification cards – Integrated circuit cards Part 4," 2013. [Online]. Available: https://www.iso.org/standard/54550.html.

[17] C. Research, "SEC 1: Elliptic Curve Cryptography," 2000. [Online]. Available: http://www.secg.org/SEC1-Ver-1.0.pdf.

[18] "ISO/IEC 15408 Evaluation Criteria for IT security," 2009. [Online]. Available: https://www.iso.org/standard/50341.html.

[19] "ISO/IEC 14443-2 Identification cards - Contactless integrated circuit cards - Proximity cards - Part 2: Radio frequency power and signal interface," 2016. [Online]. Available: https://www.iso.org/standard/66288.html.

[20] "ISO/IEC 14443-4 Cards and security devices for personal identification - Contactless proximity objects -," 2018. [Online]. Available: https://www.iso.org/standard/73599.html.

[21] Certicom Research, "SEC 2: Recommended Elliptic Curve Paramters," 2010. [Online]. Available: http://www.secg.org/sec2-v2.pdf.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2019-03-10

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.