Ledger Nano S Security Target

Release 1.2

Oct 18, 2018

CONTENTS:

Security Target Identification

Identification	Ledger Nano S Security Target
Release	1.2
Date	2018-10-18
Diffusion	Public

Security Target History

Version	Date	Author	Role	Comments
1.0	2018-07-27	Alain DESTRES	Security Certification Engineer	Initial Version
1.1	2018-10-04	Alain DESTRES	Security Certification Engineer	Add clarifications
1.2	2018-10-18	Alain DESTRES	Security Certification Engineer	Add clarifications

Security Target Review

Date	Release	Reviewer	Role
2018-07-20	1.0	Charles GUILLEMET	Chief Security Officer
2018-07-23	1.0	Pierre OSDOIT	Marketing Manager in Marketing & Communication
2018-10-01	1.1	Charles GUILLEMET	Chief Security Officer
2018-10-15	1.2	Charles GUILLEMET	Chief Security Officer

CHAPTER

ONE

INTRODUCTION

1.1 Acronym

AES	**A**dvanced **E**ncryption **S**tandard
API	**A**pplication **P**rogramming **I**nterface
ANSSI	**A**gence **N**ationale de la **S**écurité des **S**ystèmes d'**I**nformation
BIP	**B**itcoin **I**mprovement **P**roposal
BOLOS	**B**lockchain **O**pen **L**edger **O**perating **S**ystem
CC	**C**ommon **C**riteria
DES	**D**ata **E**ncryption **S**tandard
EC	**E**lliptic **C**urve
ECDSA	**E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm
ECDH	**E**lliptic-**C**urve **D**iffie-**H**ellman
FIDO	**F**ast **ID**entity **O**nline
GPIO	**G**eneral **P**urpose **I**nput **O**utput
GUI	**G**raphical **U**ser **I**nterface
HSM	**H**ardware **S**ecurity **M**odule
HTTPS	**H**yper**T**ext **T**ransfert **P**rotocol **S**ecure
IC	**I**ntegrated **C**ircuit
MCU	**M**icro **C**ontroller **U**nit
Nonce	**N**umber used **once**
OLED	**O**rganic **L**ight **E**mitting **D**iode
PIN	**P**ersonnal **I**dentification **N**umber
PKI	**P**ublic **K**ey **I**nfrastructure
PSD	**P**ersonnal **S**ecurity **D**evice (synonym for the Ledger Nano S)
RGS	**R**éférentiel **G**énéral de **S**écurité
RSA	**R**ivest **S**hamir **A**delman
SE	**S**ecure **E**lement
SEPROXYHAL	**S**ecure **E**lement **PROXY** **H**ardware **A**bstract **L**ayer
SEC	**S**tandards for **E**fficient **C**ryptography
SF	**S**ecurity **F**unctions
SHA	**S**ecure **H**ash **A**lgorithm
SPI	**S**erial **P**eripheral **I**nterface
ToE	**T**arget **o**f **E**valuation
TRNG	**T**rue **R**andom **N**umber **G**enerator
U2F	**U**niversal **2** (Second) **F**actor
UM	**U**ser **M**anual
USB	**U**niversal **S**erial **B**us

Continued on next page

Table 1 – continued from previous page

UX	**U**ser e**X**perience

1.2 Terminology

Adversary	Person trying to compromise the Ledger Nano S
Attestation	One of the core security features developed by Ledger to prove the Ledger Nano S is genuine. The attestation mechanism implementation relies on a set of cryptographic protocols based on Elliptic Curve
BOLOS	The open native Operating System developed by Ledger. One of BOLOS's features is to manage Apps (delete, install) while the Ledger Nano S has already been issued on the field. This capability offering a great flexibility allows to enrich the Ledger Nano S experience.
Blockchain	A list of blocks which are all linked together and validated via a consensus mechanism
Companion App	Ledger Live (or third-party like Mycelium, MyEtherWallet, Coinomi) running in the Host to support the Legder Nano S services. For instance, the Bitcoin application, included in the Companion App displays accounts, balance, last transactions... The Companion Apps can be either desktop/laptop or smartphone oriented.
Consent	The Ledger Nano S security design is strengthened by the End-User. As soon as a sensitive operation is required, the End-User must confirm the operation via the 2 buttons
Crypto Asset	One of the digital asset whose value is saved on the blockchain
Crypto Asset address	It is a public address provided by the End-User to transfer crypto assets. This address is derived from the Public Key.
Device App	Software running in the SE on top of the BOLOS. These device Apps can be either developed by Ledger or a third-party. A Device App offers a service.
End-User	Happy owner of a Ledger Nano S. End-User is defined by general public.
Firmware	Software running on top of an hardware (both MCU -SEPROXYHAL- and SE -BOLOS)
Hardware Wallet	Physical wallet leveraging an hardware to secure sensitive assets and sensitive operations
Host	End-User machine (laptop, desktop, smartphone and tablet) running a Companion App
Key Pair	Includes both a Private Key and a Public Key
Nano S	State-of-the-art device designed, developed and manufactured by Ledger offering a set of secure services. In this Security Target, **P**ersonal **S**ecurity **D**evice (**PSD**) means Nano S.
NESCRYPT	Coprocessor for public key cryptography algorithm embedded in [ST31H320]. Ledger leverages NESCRYPT to perform some operations on the elliptic curve.
On-boarding	Set of operations (seed generation, PIN configuration...) performed during the initialization of the Ledger Nano S
Private Key	Set of secret data involved for signing a transaction under the End-User Control
Public Key	Set of data, generated from the private key, which can be distributed
SE Firmware	The SE firmware is composed of: BOLOS OS & BOLOS UX Dashboard Device App
secp256k1	Elliptic Curve defined by Certicom Research in Standards for Efficient Cryptography ([SEC_2])
Secure Element	A Secure Element is composed of a secure IC and a Secure Software
Secure IC	It is an hardware embedding a set of physical security countermeasures. The Secure IC including in the Ledger Nano S is Common Criteria certified [ST31H320CCCertificate].
Secure Software	It is a software embedding a set of logical security countermeasures. In the Ledger Nano S, Ledger has developed BOLOS and a set of Device Apps for the Ledger Nano S.
Seed	Set of data located at the top of a hierarchical tree
SEPROXYHAL	Firmware name running on top of [ST31H320]
Service	Crypto asset management, Password Manager, Second Factor Authentication are typical services offered by the Ledger Nano S
Wallet	Solution to manage your crypto assets
Wallet Type	There are 2 types of wallet: non-deterministic wallet and deterministic wallet

1.3 ANSSI References

References	Title	Version	Date
[RGS_B1]	Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques	2.03	2014-02-21
[RGS_B2]	Règles et recommandations concernant la gestion des clés utilisées dans les mécanismes cryptographiques	2.0	2012-06-08
[RGS_B3]	Règles et recommandations concernant les mécanismes d'authentification	1.0	2010-01-13
[ST31_CC]	Rapport de maintenance ST31H320 A02 Référence: ANSSI-CC-2015/59-M01 Evaluation Assurance Level: 5+ [ST31H320CCCertificate]		2016-04-20

1.4 Bitcoin Improvement Proposal References

References	Title	Date
[BIP32]	Hierarchical Deterministic wallets	2012-02-11
[BIP39]	Mnemonic code for generating deterministic keys	2013-09-10
[BIP44]	Multi-Account Hierarchy for deterministic Wallets	2014-04-24

1.5 Additional References

References	Title	Version	Date
[SEC_2]	Certicom Research Standards for Efficient Cryptography SEC 2: Recommended Elliptic Curve Domain Parameters	2.0	2010-01-27
[AIS31]	Functionality classes and evaluation methodology for physical random number generators - BSI	1	2001-09-25
[UM]	Ledger - User Manual - Ledger Nano S [Ledger_Nano_S_User_Manual]	1.0	2018-07-19

1.6 STMicroelectronics Main Hardware References

Type	Reference	Role
SE	[ST31H320]	Main Hardware offering an EAL 5+ security level as stated in [ST31H320CCCertificate]
MCU	[STM32F042K6]	Supporting Hardware

1.7 BOLOS Python Loader

Ledger has developed a dedicated tool, called BOLOS Python Loader, to communicate with the Ledger Nano S. This tool allows to perform a set of functions. Some of them are:

1. checkGenuine

This script performs a mutual authentication between the Ledger HSM and the PSD. Firstly, the PSD ensures that the HSM is genuine, then the HSM ensures that the PSD is genuine.

2. genCAPair

   This script generates a Certification Authority key pair (elliptic curve secp256k1) that will be used to perform a mutual authentication.

3. deleteApp / listApps / signApp

   As Ledger offers the opportunity to develop some Apps, these scripts aim at managing the Apps developed by a third-party.

All the functions and further details regarding the BOLOS Python Loader can be found:

1. [Python_Loader_Installation]
2. [Python_Loader_Exploitation]

1.8 Ledger Technology Details

Some additional technical details regarding the technology created by Ledger can be found in the following list:

1. [Ledger]
2. [Readthedocs]
3. [GitHubLedgerHQ]

CHAPTER

TWO

LEDGER NANO S

2.1 Operational Environment

Ledger offers a full ecosystem to interface with the dedicated services included in the cloud, offering a smooth User Experience:

1. The Ledger's secure servers (based on HSM technology) ensure the Ledger Nano S is a genuine one, proving that the Ledger Nano S is issued by Ledger
2. The optional Companion App shares the account details and connects to the corresponding blockchain network
3. The Ledger Nano S device is leveraged to perform sensitive operations (generating seed, signing transactions, submitting passwords. . .)

The diagram below illustrates the main interactions between elements when the Companion App is required:

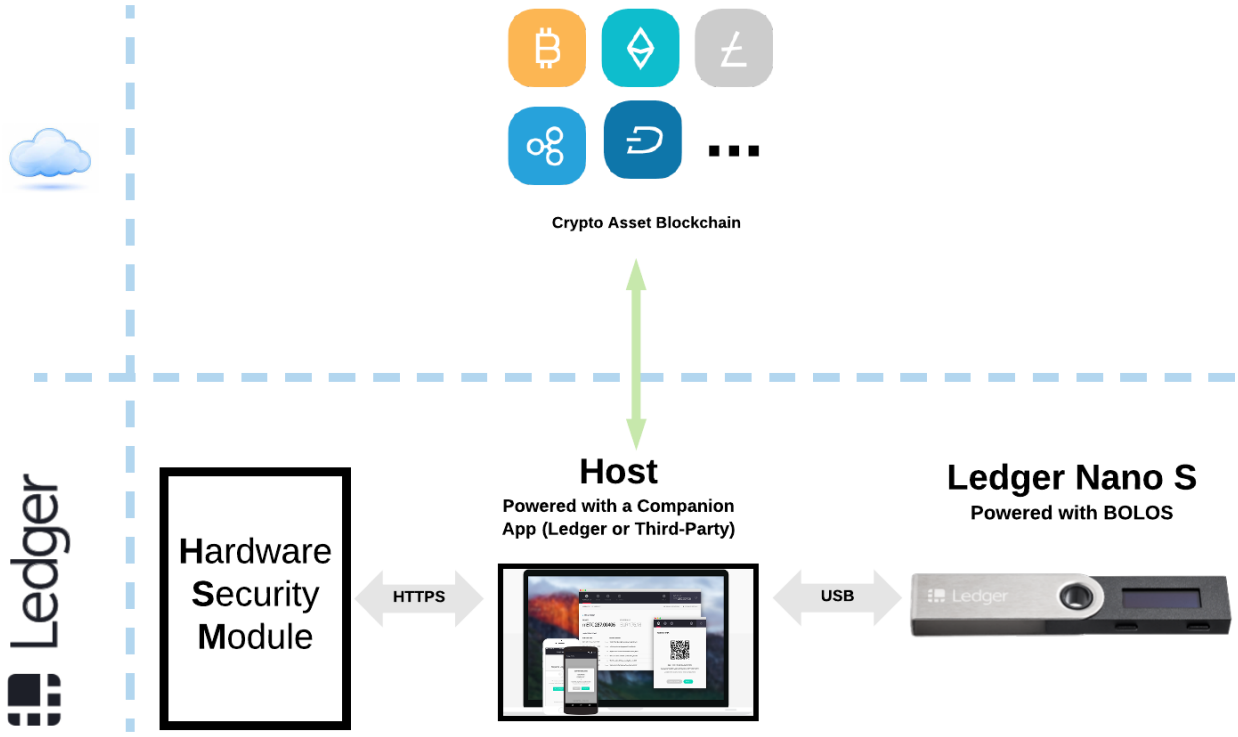

Fig. 1: **Environment WITH a Companion App**

The following diagram illustrates the main interactions between elements when the Companion App is not required:

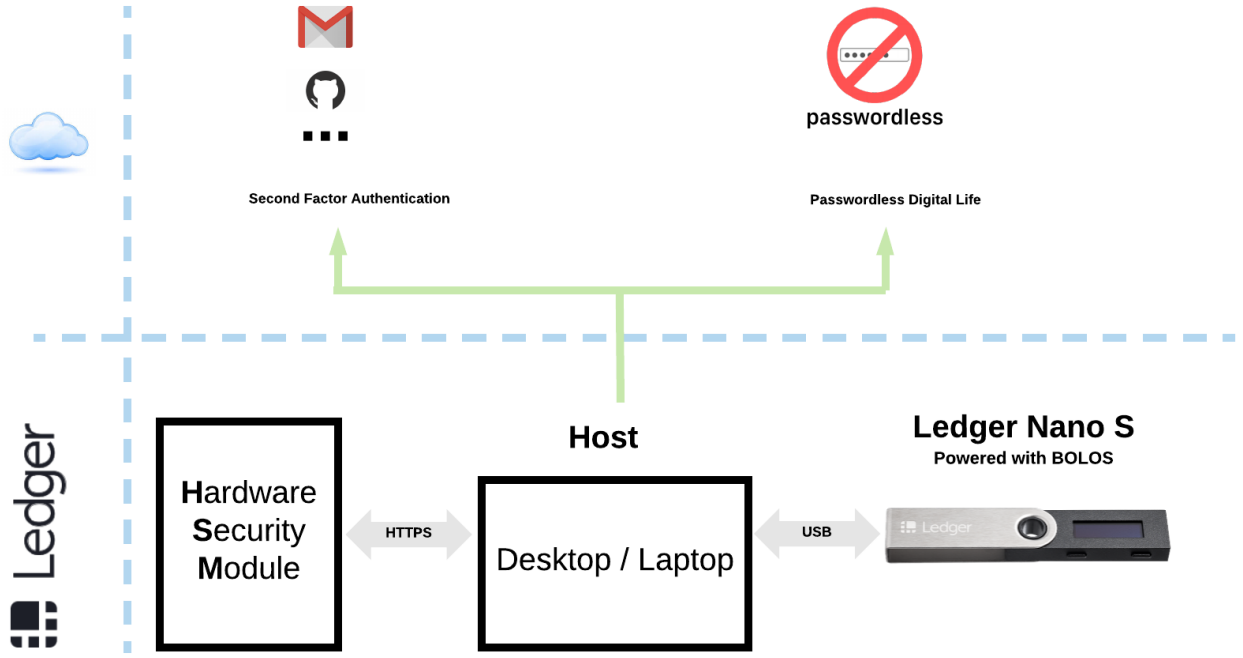

Fig. 2: **Environment WITHOUT a Companion App**

Thus, according to the use case, the End-User interacts with the:

1. Ledger Nano S (managing sensitive operations)
2. Companion App (processing non-sensitive operations) if required

2.2 Features

The Ledger Nano S supports the following features:

1. Multi-services: Hardware Wallet, Cryptographic Platform, Password Manager, Second Factor authenticator (FIDO)
2. Comply with several cryptocurrencies: Bitcoin, Bitcoin Cash, Bitcoin Gold, Ethereum, Ethereum Classic, Ethereum tokens...
3. USB connectivity
4. Open Source Device App: all Device Apps developed by Ledger can be reviewed and verified by End-Users (Bitcoin, Ethereum...)
5. Developer friendly: build a Device App and then install it on the Ledger Nano S
6. Comply with the main BIP standards: [BIP32], [BIP39] and [BIP44]
7. Multi-platform: Windows (7+), Mac (10.9+), Linux or Chrome OS
8. **OLED screen: to verify the transaction data (amount, address)**
9. **Buttons: used to get the End-User's consent relative to sensitive operations like unlocking the device or processing a transaction**
10. **PIN: to unlock the Ledger Nano S**

11. **Plausible deniability: an additional PIN linked to a passphrase can be defined to create an hidden account**
12. **Genuine: sophisticated attestation mechanisms ensuring that the Ledger Nano S is a genuine one**
13. **Post-issuance capability: all piece of software (MCU Firmware, SE Firmware, Device Apps) can be securely updated**

Bold features are included in the security scope and addressed by dedicated security functions.

2.3 Services

Services are not included in the security scope. These services are not addressed in the scope because they are all protected by the End-User's PIN. Indeed, the Ledger Nano S requires the End-User's PIN unlocking then all services listed in the following sections. Thus, even if the services are out of scope, the secret data belonging to services are properly protected through the PIN.

2.3.1 Hardware Wallet Service

The wallet is the main service.

It is the combination of the following two elements that creates an operational wallet:

1. Companion App executed on the Host
2. Ledger Nano S with the dedicated crypto asset application installed and currently selected. The Ledger Nano S acts as a secure gateway to the blockchain technology.

This wallet service managing crypto assets is in charge of:

1. Managing the balance (Companion App)
2. Handling one or several accounts (Companion App)
3. Supporting one or several crypto assets: Bitcoin, Bitcoin Cash, Bitcoin Gold, Ethereum, Ethereum Classic… (Companion App & Ledger Nano S)
4. Processing transactions: receive & perform payments (Companion App & Ledger Nano S)

If you remove one of these elements, no transaction can be processed. The Host performs no security operations. All sensitive operations (for instance signing a transaction, confirming the amount of the transaction, confirming the recipient's address) are directly performed with the Ledger Nano S based on the Secure Element technology. The security model designed by Ledger relies on the Ledger Nano S including not only a certified secure IC [ST31H320CCCertificate] but also a secure software developed by Ledger.

2.3.2 Cryptographic Platform Service

The Ledger Nano S, considered as a cryptographic embedded platform, supports several cryptographic primitives as listed below (not limited to):

1. Symmetric cryptography: DES/3DES, AES
2. Asymmetric cryptography: RSA (key size: 1024, 2048, 3072, 4096 bits), EC (brainpool, SECP and ANSSI)
3. Secure Hash: SHA224, SHA256, SHA384, SHA512

2.3.3 Password Manager Service

A Device App manages all your passwords making the connection step easier for an End-User.

2.3.4 FIDO Service

The FIDO U2F Device App is a two-factor authentication method specified by the FIDO Alliance. It works with several web services, like Facebook, Dashlane, Gmail, Dropbox, GitHub, etc.

For each of these web services, the End-User needs to set up the security parameters of the account to register the Ledger Nano S as a second factor security key to authenticate on it. This second factor of verification will improve the security of your log in processes, as the End-User will be first required login/password followed by the second factor via the Ledger Nano S.

2.3.5 Additional Innovative Services

As the Ledger's ecosystem is developer-friendly, a third-party can develop a Device App to build an innovative and useful service.

2.4 Dual Architecture

The Ledger Nano S is based on an architecture leveraging two hardware:

1. a generic MCU: [STM32F042K6]
2. a Secure Element: [ST31H320]

The [STM32F042K6] can be considered as a supporting hardware and is in charge of:

1. Managing the USB communication with the Host
2. Driving the screen
3. Receiving the notifications from the buttons
4. Communicating with the SE

The [ST31H320], as it belongs to the Secure Element Technology and is Common Criteria certified (refer to [ST31H320CCCertificate] to get further details), ensures all sensitive operations and is in charge of (but not limited to):

1. Generating the seed
2. Deriving the corresponding Key Pair
3. Signing transactions
4. Communicating with the MCU

Note that Ledger Nano S can be used without a Companion App. Indeed, both Password Manager and FIDO Device Apps directly connect to the web service without a Companion App.

The Ledger Nano S relies on the Secure Element technology addressing the security issues linked to the storage and manipulation of secret keys. The Secure Element technology is leveraged in sensitive applications: for instance banking card, passport, driving licence. The Ledger Nano S also leverages this Secure Element technology to protect properly the End-User's assets.

The Ledger Nano S is a secure hardware wallet thanks to the proper exploitation of the Secure Element technology. The Ledger Nano S can be used for instance in relation with a Companion App running on top of the Host as illustrated by the following diagram:

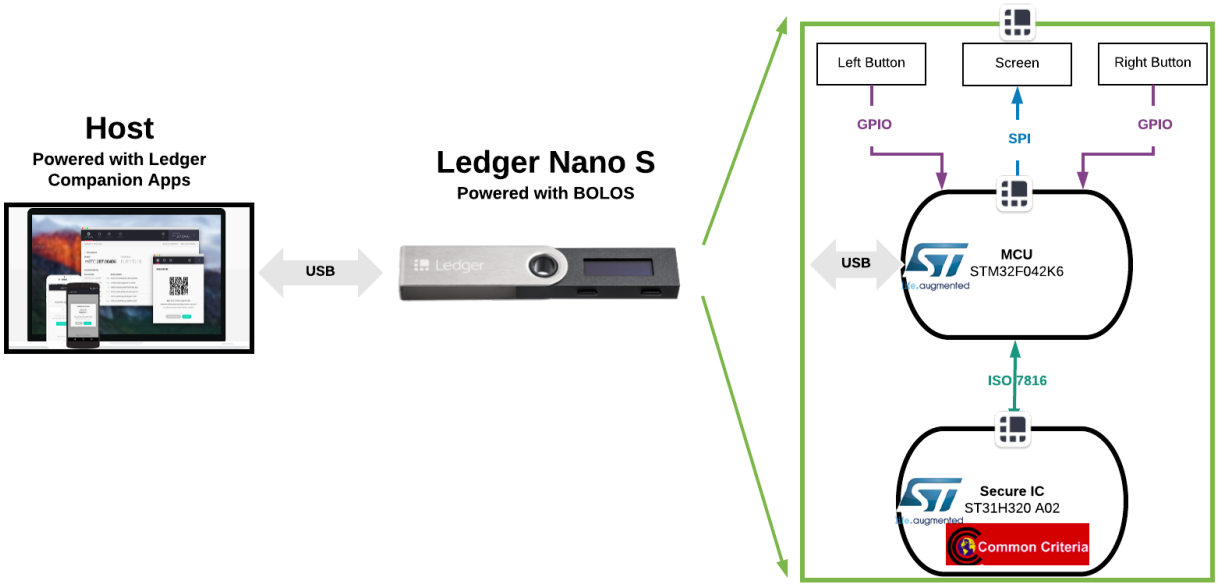

Fig. 3: **Zoom on the Ledger Nano S**

Besides, the End-User actively participates in the security: all sensitive operations must get the End-User's consent (PIN validation, transaction confirmation) achieved via the screen and buttons.

One of the main advantages of the dual architecture is the fact that all modules around the Ledger Nano S can be hacked without compromising the overall security. For instance, if the Host is compromised, it cannot introduce a security breach in the Ledger Nano S. The security model defined by Ledger is that only the Ledger Nano S can be trusted. Thus, this solution is resistant against threats like malware proof.

2.5 Identification

The following table identifies the Ledger Nano S:

Developer name	Ledger SAS, 1 rue du Mail, 75002 Paris
Visit our website	http://www.ledger.com
Product Type	Personal Security Device
Product Category	Hardware and Embedded Software
Product name	Ledger Nano S
Reference	2c970001
SE Firmware Version (BOLOS)	1.5.1 [ST31H320]
MCU Firmware Version (SEPROXYHAL)	1.6 [STM32F042K6]

The product identification can be directly processed by the End-User on the Ledger Nano S. The following steps must be performed:

1. Connect the Ledger Nano S to the Host
2. Enter and validate the PIN

3. Select “Settings”, “Device” and “Firmware” menu
4. Verify that the version (Secure Element and MCU) displayed on the screen are identical to the ones identified in the previous table.

2.6 Target of Evaluation

The Personal Security Device is an embedded platform processing securely sensitive services. The PSD includes a set of core security mechanisms (TRNG, End-User verification via the enrolled PIN, attestation mechanism, post-issuance capability). These security mechanisms linked with a simplified User Experience makes the PSD usage secured and simple.

The security model created by Ledger is based on the Secure Element technology. This Secure Element embeds a set of hardware security countermeasures (for instance active shield, monitoring of environmental parameters, True Random Number Generator).

Nevertheless, in order to get a product resistant against high attack potential, Ledger has also implemented a set of software security countermeasures. It is the composition of hardware security mechanisms (provided by the Secure IC) and the software security mechanisms (provided by Ledger) which make the Ledger Nano S resistant against sophisticated attacks (elapsed time, expertise, equipment).

The Target of Evaluation, focused on the Ledger Nano S, is identified in the following diagram:

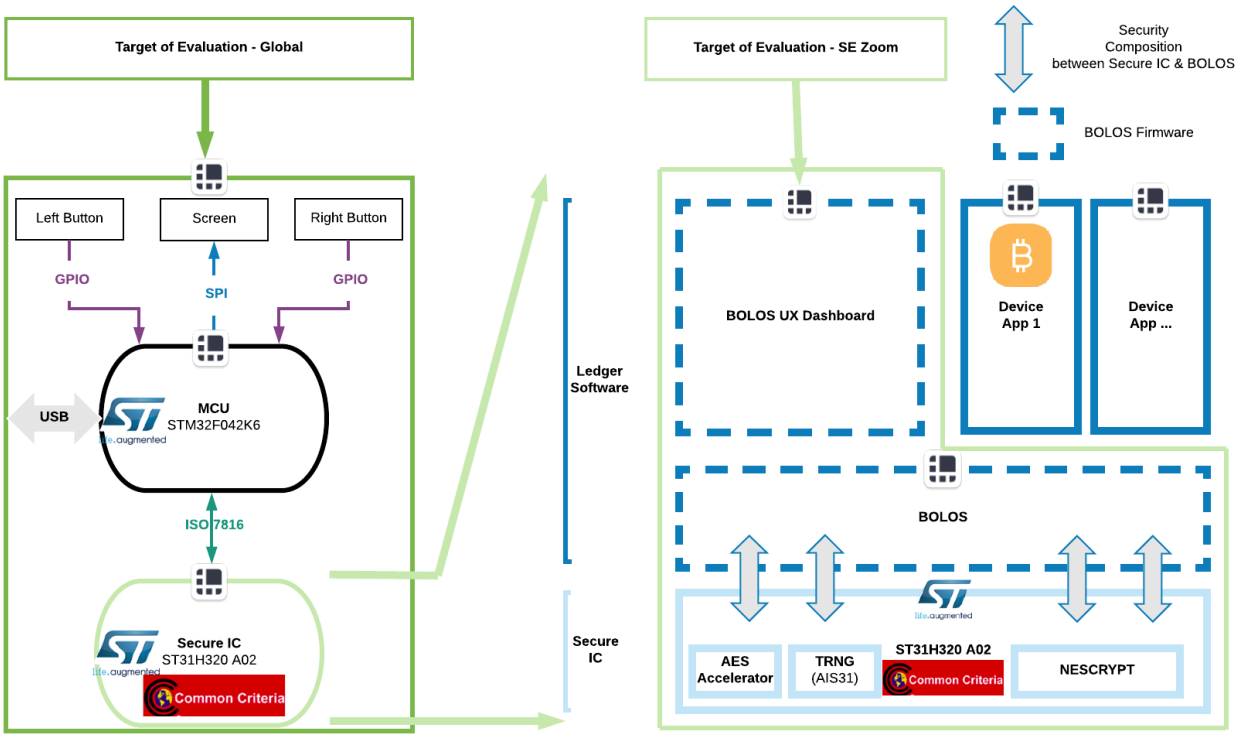

Fig. 4: **Target of Evaluation including a zoom on the SE**

The ToE includes:

1. **Physical elements**
    (a) Two buttons

   (b) One screen

2. **Hardware (provided by STMicroelectronics)**
   (a) MCU: [STM32F042K6]
   (b) Secure IC: [ST31H320] (Common Criteria certified)
3. **Software (developed and secured by Ledger)**
   (a) SEPROXYHAL firmware running on top of [STM32F042K6]
   (b) **BOLOS firmware running on top of [ST31H320] contains:**
      i. an OS labelled BOLOS
      ii. a Device App labelled BOLOS UX Dashboard

BOLOS is in charge of:

1. Communicating with the outside world
2. Performing cryptographic computation
3. Storing secret data (seed, PIN)
4. Offering a set of API (communication, cryptographic primitives, seed) accessible to all Device Apps

The BOLOS UX Dashboard Device App, default Device App active as soon as the PIN is successfully verified, is:

1. the entry point to select another Device App
2. in charge of the on-boarding phase: seed generation and PIN enrollment
3. involved in the other Device App management (delete, install)

The BOLOS UX Dashboard Device App ensures a consistency UX whatever the running Device App. This Device App manages for instance buttons and the screen. Thus, this Device App also supports a third-party developer to create his own Device App.

All Device Apps (developed by Ledger or not), except BOLOS UX Dashboard Device App are not included in the ToE.

2.7 Assumptions

Below is the list of assumptions:

1. The Ledger Nano S is acquired from an official Ledger reseller (Ledger, Amazon stores)
2. The HSM is properly operated by Ledger
3. The End-User has verified that the Ledger Nano S has not been tampered ([CheckHardwareIntegrity])
4. The End-User only installs non-malicious Device Apps

2.8 Environment Measures

Even if the Ledger Nano S can be used within a strict environment (for instance storing the device inside a vault, signing a transaction inside a secure building), the security design developed by Ledger allows the End-User to experience the PSD in a public area. The device is architectured to provide an high assurance level to the End-User whatever the environment.

Nevertheless, with the Ledger Nano S security model, the security is shared between the Ledger Nano S device and the End-User. The following security rules must be fulfilled:

1. The End-User **must** maintain the PIN secret
2. The End-User **must** enter the PIN through the 2 buttons and the screen so that it is impossible to an adversary to get the PIN value
3. The End-User **must** keep the recovery sheet secret
4. The End-User **must** ensure that no one has access to the recovery sheet
5. The End-User **must** verify transactions' details and address displayed on the screen are valid

2.9 End-User

One of Ledger's ambitions is to ensure the blockchain technology can be appropriated by all. While the solution is sophisticated, the Ledger Nano S offers a simple and natural User Experience. The Ledger Nano S being designed to be a mainstream technology, it is user-friendly and can be easily manipulated by End-Users with no technical background.

CHAPTER

THREE

ASSETS

As the PSD processes sensitive operations (i.e. sign transactions, manage passwords, achieve U2F authentication, . . .) and stores confidential data, the following primary assets must be secured:

1. Random number - data
2. Secret seed - data
3. Secret Data (protected by the PIN) - data
4. PSD Access Control - operation
5. SE Firmware - data
6. MCU Firmware - data

All the primary assets listed above is worth of interest to an adversary and are subject to a set of threats as mentioned in *Threats*.

CHAPTER

FOUR

THREATS

4.1 Threat Agent

The Ledger Nano S can be considered as a sensitive device: for instance, it can be used to process some sensitive operations related to the management of the corresponding crypto assets belonging to the End-User.

One of the Ledger Nano S features is to sign digital transactions. This signature operation, used to unlock the cryptocurrency funds located on the blockchain, involves the manipulation of the private key. There is a link between the owner of this private key (the End-User) and the owner of the cryptocurrency funds.

Additionally, the Ledger Nano S is not only an hardware wallet but also provides a set of added-value services. The Password Manager and FIDO are typical Apps making the digital End-User life frictionless and more secured.

Several threats are applicable to the Ledger Nano S and can be divided into two classes:

1. Physical threats
2. Remote threats

With the physical threats class, the adversary has a physical access to the Ledger Nano S. This occurs when the Ledger Nano S has been either stolen or lost.

As the Ledger Nano S is a device to support sensitive online services (blockchain, second factor authentication, passwordless), the remote threats class is also applicable. With this class, the adversary has no physical access. This remote threats class is considered when the End-User's Host has been compromised. It is through this infected machine (desktop, laptop, smartphone, tablet) that an adversary will launch his attacks (i.e. signing a transaction, getting passwords).

The following section describes the main threats linked to the Ledger Nano S related to the two threat classes.

4.2 Threat #1: Generating a biased or a deterministic random number

4.2.1 Context

The Random Number Generator included in the Ledger Nano S is used to:

1. Generate a Random Number exploited as a seed
2. Participate in establishing a secure channel between the Ledger Nano S and the the Ledger's HSM

The Ledger Nano S, compliant with [BIP32], is a deterministic hardware Wallet. This feature indicates that a seed is generated by the device during the initialization. From this seed, the End-User has the capability to derive all Key Pairs required to manage the crypto assets accounts.

Note that this feature allows to recover the crypto assets funds if the Ledger Nano S is lost, stolen or destroyed as long as the seed is correctly backed up.

The Ledger Nano S uses the Random Number Generator not only for generating the seed but also for creating a Secure Channel between the Ledger's Secure Server and the Ledger Nano S to avoid replay attacks.

4.2.2 Threat

The entropy is the key element regarding a Random Number feature. The entropy must be ensured by a true random number. The main threat is to reduce the entropy so that it reduces dramatically the seed space. This seed's space size is 2^{256}.

Another threat is to generate a number under the control of an adversary. This number is leveraged to set-up the End-User's account by creating the crypto address. Then, the End-User's account can be provisionned.

As the number generation is controlled by the adversary, it is then possible to recreate the End-User's account and perform a crypto asset transfer from the End-User's account to the adversary's account.

4.3 Threat #2: Using a not genuine Ledger Nano S

4.3.1 Context

Ledger is the only manufacturer of the Ledger Nano S device. The authenticity proves the Ledger Nano S is only issued by Ledger avoiding some security holes. Besides, as the Ledger Nano S is a sensitive device, it must only work as specified by Ledger. For instance, the Ledger embedded software, including not only BOLOS and but also a set of Device Apps, must be executed as expected.

In other words, both authenticity and integrity of the Ledger Nano S must be ensured.

4.3.2 Threat

The main threats are:

1. Manufacturing a fake Ledger Nano S

   As an adversary manufactures a fake Ledger Nano S, it has the full control on the device and can create malicious Ledger Nano S.

2. Modifying the Ledger Nano S produced by Ledger

   An adversary adds malicious software to dump out sensitive data.

4.4 Threat #3: Bypassing the Access Control to Sensitive Services

4.4.1 Context

The Ledger Nano S embeds a set of sensitive services. One of them is related to the management of crypto assets. For instance, an End-User can create on his Ledger Nano S a set of accounts linked to several cryptocurrencies (Bitcoin, Ethereum, Ripple). The Apps installed on the Ledger Nano S can sign transactions to unlock the funds.

Note that the Ledger Nano S offers several sensitive services (FIDO, Password Manager) which can be interesting for an adversary as well.

4.4.2 Threat

The main threat is related to a stolen Ledger Nano S. As soon as the Ledger Nano S is stolen, the adversary gets a full control over the device meaning that sensitive operations can be processed.

This threat is also applicable remotely when the End-User's Host has been previously compromised.

4.5 Threat #4: Compromising the Post-Issuance Capability

4.5.1 Context

The Ledger Nano S includes a post-issuance capability making possible to update not only Device Apps but also firmwares (both BOLOS and SEPROXYHAL). This feature, giving Ledger an incredible flexibility, can be exploited to:

1. add new services
2. fix some functional and protocol issues
3. reinforce the security of the Ledger Nano S

Finally, the update is either a firmware or a Device App.

4.5.2 Threat

The main threat is to inject a malicious firmware (either SE or MCU) so that an adversary can take the full control.

CHAPTER

FIVE

SECURITY FUNCTIONS

As raised in the previous section, the Ledger Nano S can be targeted with four main threats. These threats are critical because they can compromise the Ledger Nano S: deterministic random number, access to the device without End-User verification, fake or malicious Ledger Nano S.

Ledger has developed appropriate security functions explained in this section to properly block each threat. It is worth highlighting that the implementation of these security functions relies on a set of security mechanisms. This security methodology of adding several security layers counteracts not only straightforward but also sophisticated attacks.

The diagram below illustrates the relationship between the threats and the security functions:

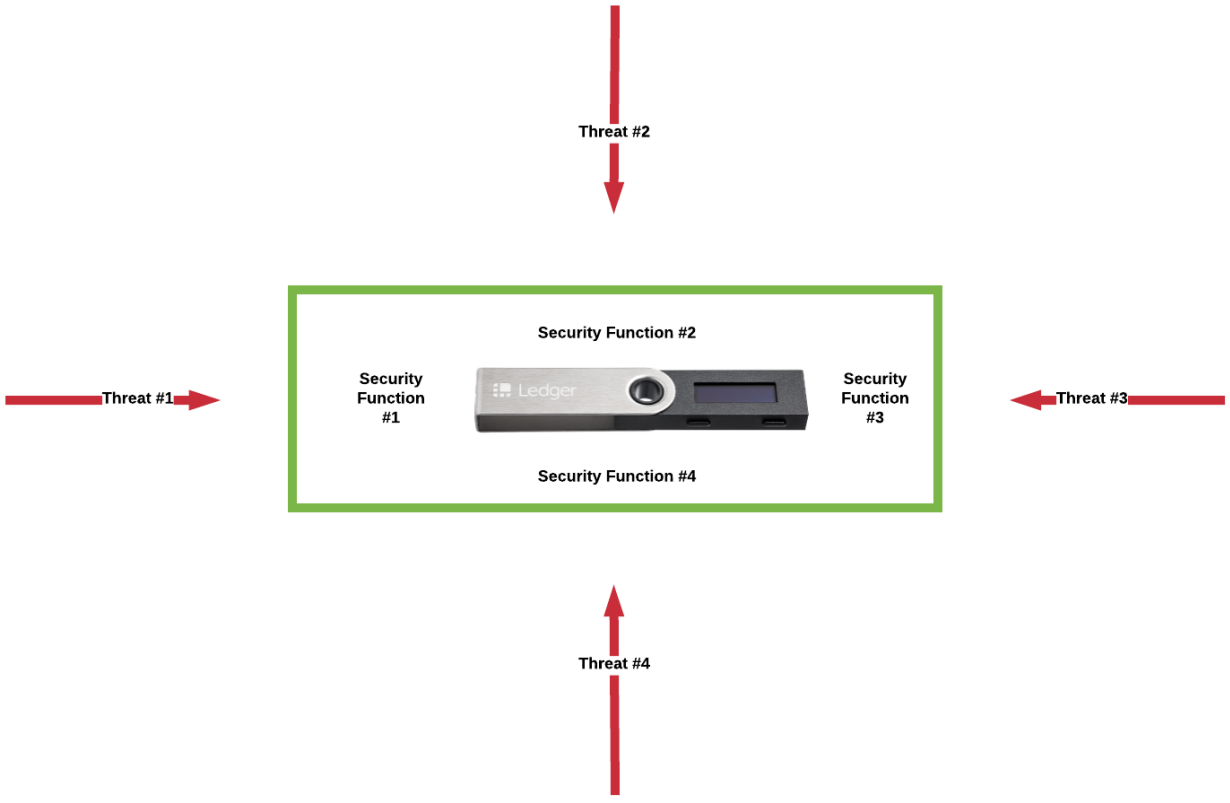

Fig. 1: **Threats Vs Security Functions**

5.1 Security Function #1: True Random Number Generator

5.1.1 Description

This security function #1, labelled True Random Number Generator, aims at counteracting threat #1.

This security function is based on the TRNG embedded in [ST31H320]. This TRNG, evaluated according to [AIS31] methodology, has been successfully certified Class PTG.2.

To reinforce the entropy of the generated Random Number, Ledger has also implemented an additional software post-processing countermeasure.

5.1.1.1 Assets

The assets related to security function #1 are:

1. True Random Number Generator (operation)
2. Random Number (data)
3. Secret Seed (data)

5.2 Security Function #2: Attestation Mechanism

5.2.1 Description

This security function #2, labelled Attestation Mechanism, aims at blocking threat #2.

Ledger has implemented a solution to ensure that the Ledger Nano S belonging to the End-User is a genuine one. To comply with this requirement, a Public Key Infrastructure (based on secp256k1 elliptic curve) has been set up: Ledger is the Certification Authority. This dedicated infrastructure, based on Hardware Security Module, is administrated by Ledger.

During the manufacturing process, each Ledger Nano S initializes itself with an individual key pair generation and the corresponding certificate (provided by the Ledger's HSM). Then, when the Ledger Nano S is connected to the Host and under some circumstances (for instance a Device App, SE firmware download or MCU firmware installation), a mutual authentication between the Ledger's HSM and the Ledger Nano S is performed.

This security function #2 relies on the following commands:

1. VALIDATE_TARGET_ID
2. INITIALIZE_AUTHENTICATION
3. VALIDATE_CERTIFICATE_LAST
4. GET_CERTITICATE_LAST

The VALIDATE_TARGET_ID and INITIALIZE_AUTHENTICATION commands initiate the mutual authentication. The VALIDATE_CERTIFICATE_LAST command is used by the PSD to authenticate the HSM while the GET_CERTIFICATE_LAST command is used by the HSM to authenticate the PSD.

At the end of this command/response sequence, a mutual authentication is achieved. Besides, both HSM and PSD have generated ephemeral keys leveraged during an ECDH to share a common secret between the HSM and the PSD.

This attestation mechanism is performed through a set of ECDSA operations as illustrated in the following diagram:

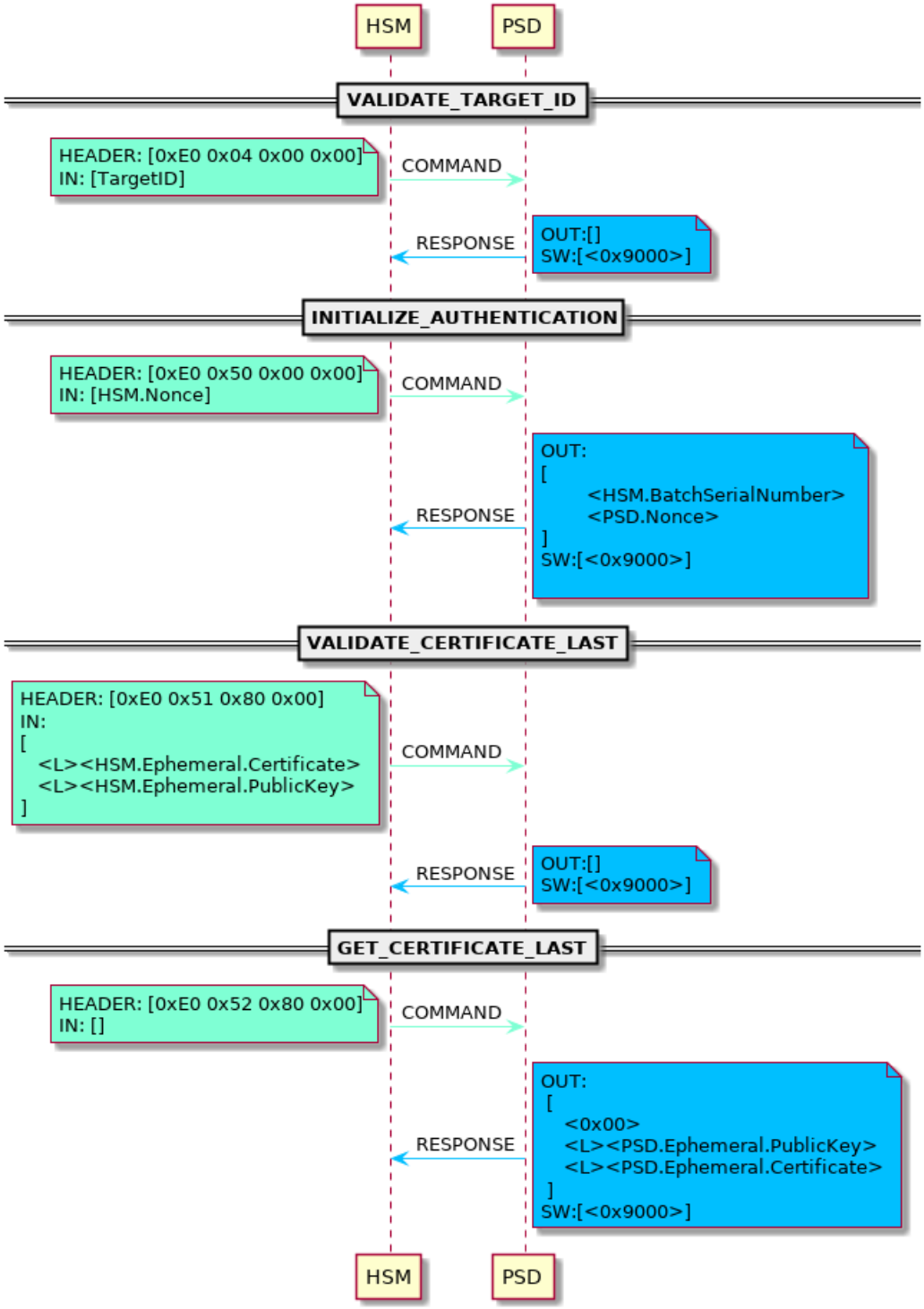

Fig. 2: **Security Function #2 - Attestation Mechanism -**

5.2.2 Assets

The assets related to security function #2 are:

1. True Random Number Generator (Operation)
2. Random Number (Data)
3. PSD Genuineness (Data)
4. PSD.PublicKey (Data)
5. PSD.Ephemeral.PrivateKey (Data)
6. PSD.PrivateKey (Data)
7. HSM.Ephemeral.PublicKey (Data)
8. HSM.Ephemeral.Certificate verification (Operation)
9. HSM.PublicKey (Data)

5.3 Security Function #3: End-User Verification

5.3.1 Description

This security function #3, labelled End-User Verification, aims at counteracting threat #3.

As soon as the Ledger Nano S is connected to a Host, the End-User must prove that he is the owner of this Ledger Nano S. This security function #3 is the first interaction between the End-User and the Ledger Nano S. This security function is critical because it gives access to all services supported by the Ledger Nano S.

The End-User verification is performed through a PIN verification. The length of the PIN, defined by the End-User during the on-boarding stage, must be in the following range: minimum 4 digits, maximum 8 digits.

The End-User directly enters the PIN value using the 2 buttons. This candidate PIN is then compared to the Reference PIN stored in the SE. A correct verification allows the End-User to use all services provided by the Ledger Nano S. For instance, all cryptocurrency Apps are available meaning cryptocurrency transfer is available. Note that all other Apps (for instance Password Manager, FIDO) are also available as soon as the PIN verification is successfully performed.

The PIN Try Counter, whose default value is set to 3, counteracts brute-force attacks revealing the value of the PIN. As soon as the PTC exceeds its limit, the Ledger Nano S wipes the following sensitive assets:

1. PIN
2. Seed
3. Secret Data protected by the PIN

Thanks to this security action of wiping, the Ledger Nano S cannot be used because the current state is not operational anymore. An initialization (either normal mode or restore mode) is then required.

A correct End-User verification unlocks all the Ledger Nano S services and resets the PTC.

5.3.2 Assets

Several sensitive assets are used to ensure the End-User verification:

1. PSD Access Control (Data)
2. Secret data protected by the PIN (Data)

3. Reference PIN (Data)
4. PIN Try Counter (Data)
5. PIN verification (Operation)
6. PIN Try Limit (Data)
7. PIN Result (Data)

5.4 Security Function #4: Post-Issuance Capability over a Secure Channel

5.4.1 Description

This security function #4, labelled Post-Issuance Capability over a Secure Channel, aims at counteracting threat #4.

This security function use security assets generated during security function #2 execution (mutual authentication between the HSM and the PSD).

The Post-Issuance Capability over a Secure Channel is illustrated in the following diagram:

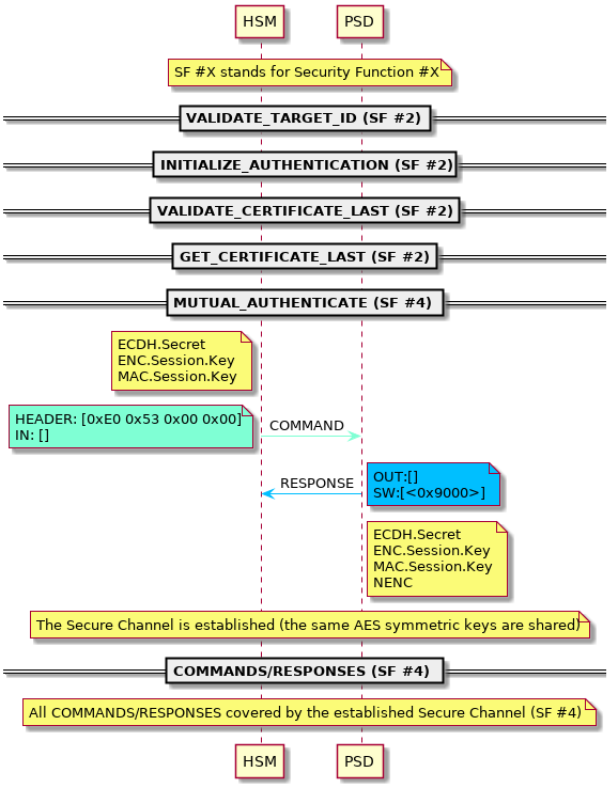

Fig. 3: **Security Function #4 - Secure Channel -**

The first commands (VALIDATE_TARGET_ID, INITIALIZE_AUTHENTICATION, VALIDATE_CERTIFICATE_LAST, GET_CERTIFICATE_LAST) performs a mutual authentication (security function #2) to ensure the HSM and PSD are genuine. Note that during the execution of the previous commands, both HSM and PSD have generated ephemeral EC key pairs. These ephemeral key pairs are leveraged to process an ECDH so that both HSM and PSD share a common secret labelled ECDH.Secret.

This ECDH.Secret is then derived to get 2 session keys:

1. ENC.Session.Key
2. MAC.Session.Key

These 2 session keys ensure the confidentiality and the integrity of messages (command/response) over the secure channel.

There is an additional key, labelled NENC, used to only encrypt the SE firmware. In this case, the SE firmware is encrypted twice: the first encryption is achieved through NENC while the second encryption is processed with ENC.Session.Key. NENC is provisionned to the PSD during the previous SE firmware update.

After the successful processing of the MUTUAL_AUTHENTICATE command, all following commands (secured in confidentiality and integrity) are managed inside the Secure Channel.

The secure channel is designed to block typical attacks. For instance, the secure channel does not accept the same set of commands twice making replay attacks not operational anymore. Additionaly, thanks to the NENC's use, the software installation is always an upgrade. It is not possible to downgrade the software version already installed on the Ledger Nano S. This anti-rollback security protection discards all attack vectors related to install a previous software version containing a set of vulnerabilities already identified and exploited.

5.4.2 Assets

1. MCU Firmware (Data)
2. SE Firmware (Data)
3. ECDH.Secret (Data)
4. ENC.Session.Key (Data)
5. MAC.Session.Key (Data)
6. NENC (Data)

CHAPTER

SIX

SUMMARY: THREATS - ASSETS - SECURITY FUNCTIONS

6.1 Mapping Between Assets and Security Functions

The following table gives the relationship between assets and security functions.

#	**Asset Name**	**Asset Type**	**Security Function**	**Memory Type**	**Property**
1	True Random Number Generator	Operation	#1 and #2	Not applicable	I
2	Random Number	Data	#1 and #2	Volatile	
3	Secret Data (seed)	Data	#1	Persistent	I&C
4	PSD Access Control	Data	#3	Not appllicale	AU
5	Secret Data protected by the PIN	Data	#3	Persistent	I&C
6	Reference PIN	Data	#3	Persistent	I&C
7	PIN Try Counter	Data	#3	Persistent	I
8	PIN Verification	Operation	#3	Not applicable	I
9	PIN Try Limit	Data	#3	Persistent	I
10	PIN Result	Data	#3	Volatile	I
11	PSD Genuineness	Data	#2	Not applicable	AU
12	PSD.PublicKey	Data	#2	Not applicable	I&C
13	PSD.Ephemeral.PrivateKey	Data	#2	Volatile	C
14	PSD.PrivateKey	Data	#2	Persistent	I&C
15	HSM.Ephemeral.PublicKey	Data	#2	Volatile	C
16	HSM.Ephemeral.Certificate Verification	Operation	#2	Not applicable	I
17	HSM.PublicKey	Data	#2	Volatile	I&C
18	SE Firmware	Data	#4	Volatile	I
19	MCU Firmware	Data	#4	Volatile	I
20	ECDH.Secret	Data	#4	Volatile	I&C
21	ENC.Session.key	Data	#4	Volatile	C
22	MAC.Session.key	Data	#4	Volatile	C
23	NENC	Data	#4	Persistent	C

I = **I**ntegrity **C** = **C**onfidentiality **A** = **A**vailability **AU** = **AU**thenticity

6.2 Mapping Between Security Functions and Threats

The following table gives the full relationship between the security functions and the threats.

	Threat #1	Threat #2	Threat #3	Threat #4
True Random Number Generator	X			
Attestation Mechanism	X	X		
End-User Verification			X	
Post-Issuance Capability over a Secure Channel		X		X

Threat #1 is also applicable to security function #2 because a Nonce is required during the attestation mechanism. Besides, as security function #4 is based on security function #2, Threat #2 is also applicable to security function #4.

CHAPTER

SEVEN

USE CASES

7.1 On-Boarding

The on-boarding use case is leveraged to properly initialize the Ledger Nano S. As soon as the Ledger Nano S is received by the End-User, this on-boarding is:

1. a mandatory step: without on-boarding, the Ledger Nano S is useless
2. the first step which must be completely performed before using the Ledger Nano S

To initialize the device, the End-User has to select one of the following modes:

1. Normal mode (“Initialize as new device”)
2. Restore mode (“Restore a configuration”)

The restore mode is performed when the End-User has already a wallet and wants to use the Ledger Nano S (compliant with [BIP32]) to get access of his assets again. The normal mode is used when the End-User wants to create a seed.

Whatever the selected mode, the End-User has to choose a PIN code (from 4 to 8 digits long). Within the normal mode, the 24-word recovery phrase is displayed word by word and must be written down on the recovery sheet. With the restore mode, the End-User must enter the 12/18/24-word recovery phrase saved during the initialization phase. Note that during the on-boarding phase, the Ledger Nano S only generates a recovery phrase composed of 24 words.

Refer to *On-boarding Flow*, to get the diagram describing the on-boarding process making the Ledger Nano S operational.

As soon as the initialization is completed by the End-User, the Ledger Nano S switches to the operational state.

7.2 Typical scenarios

As the Ledger Nano S is a generic personal secure device offering a comprehensive set of security features, there are as well a lot of scenario. For instance, The Ledger Nano S can be used within the following scenarios (but not limited to):

1. Install the official Bitcoin Device App (in other words, signed by Ledger)
2. Perform a Bitcoin transaction
3. Perform a FIDO transaction
4. Install the latest SE firmware
5. Install a Device App developed by the End-User (in other words not signed by Ledger)

All these typical scenarios are sensitive because they are manipulating sensitive assets. The Ledger Nano S embeds 4 security functions implemented to secure the Ledger Nano S assets.

The following table maps some scenarios with the corresponding security functions. Note that the security functions are listed in a chronological order meaning that Security Functions #3 (End-User verification) is the first step, Security Function #1 is the second step and so on for the remaining security functions.

	Security Functions			
Typical Scenario	SF #3	SF #1	SF2 #2	SF #4
Install the official Bitcoin Device App	X	X	X	X
Perform a Bitcoin transaction	X			
Perform a FIDO transaction	X			
Install the SE firmware	X	X	X	X
Install a Device App developed by the End-User	X			

As illustrated by the table above, Security Function #3 (End-User verification) is always performed whatever the scenario. It is even the first security action that the End-User's must process before using the Ledger Nano S.

As this Security Function #3 is crucial to the Ledger Nano S, all data are wiped (including PIN, Seed and all other Secret Data) as soon as the PTC exceeds its limit. Note that the End-User has then the possibility to restore the Ledger Nano S through the set of words written down the recovery sheet.

Depending on the use case, other Security Functions (#1, #2 and #4) may be invoked.

CHAPTER

EIGHT

ANNEX

8.1 On-boarding Flow

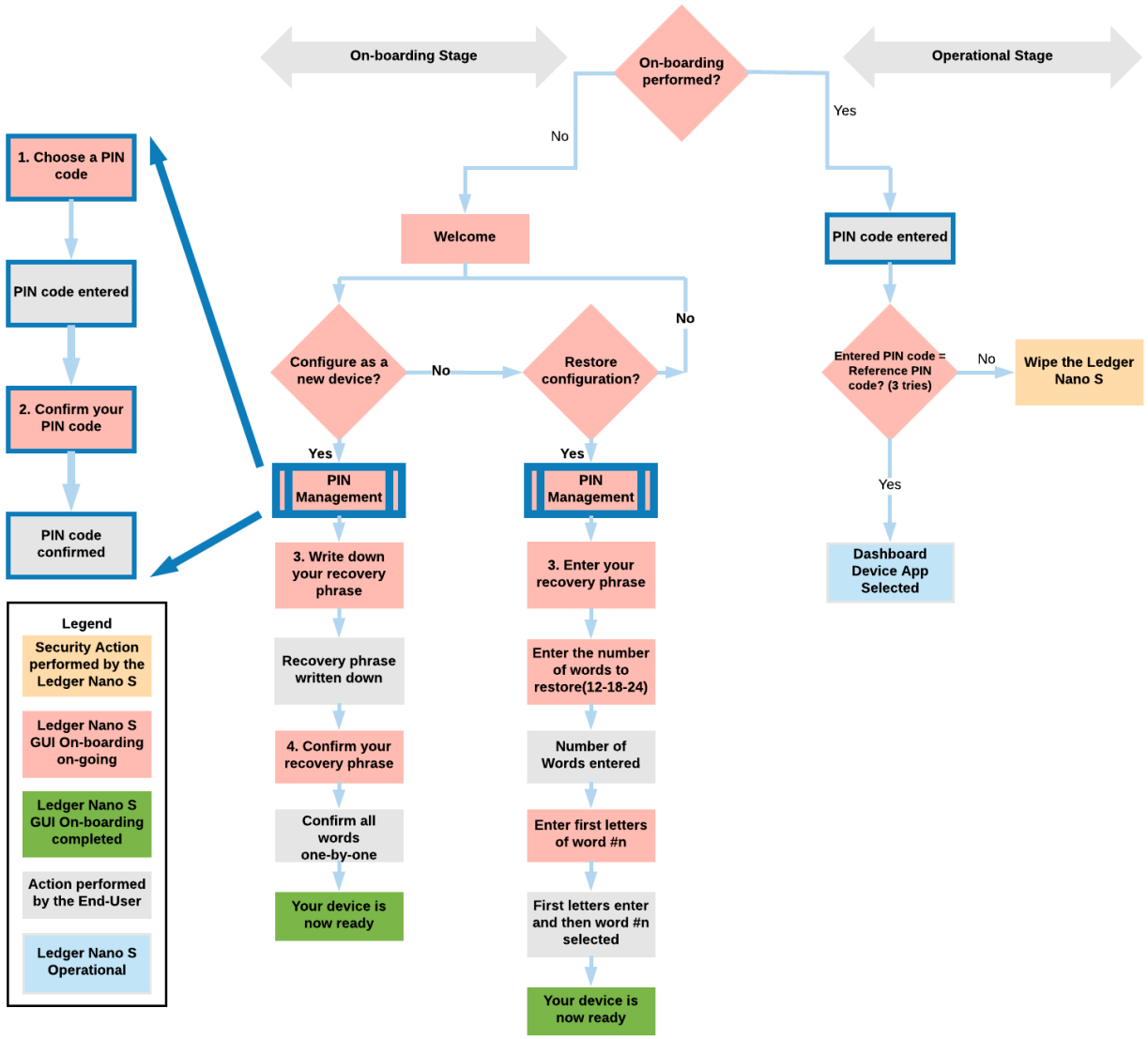

Fig. 1: **On-Boarding Flow**

8.2 External References

[ST31H320CCCertificate]	https://www.commoncriteriaportal.org/files/epfiles/ANSSI-CC-2015_59-M01.pdf
[Python_Loader_Installation]	https://github.com/LedgerHQ/blue-loader-python
[Python_loader_Exploitation]	https://ledger.readthedocs.io/projects/blue-loader-python/en/0.1.18/script_reference.html
[ST31H320]	https://www.st.com/en/secure-mcus/st31h320.html
[Ledger]	https://www.ledger.com
[Readthedocs]	http://ledger.readthedocs.io/en/latest/
[GitHubLedgerHQ]	https://github.com/LedgerHQ
[CheckHardwareIntegrity]	https://support.ledgerwallet.com/hc/en-us/articles/115005321449
[Ledger_Nano_S_User_Manual]	https://support.ledgerwallet.com/hc/en-us/articles/360007061974-User-Manual