Avixy 4000

Security Policy

Revision 1.3

January 2016

General Information

Project	Avixy 4000
Title	Security Policy
Owner	Avixy Tecnologia
Document Type	Report

Version Control

Rev.	Description	Responsible	Approved by	Date
1.3	Removed support to WEP	Gustavo Cotta	Eduardo Marques	01/2016
1.2	Included information on how to proper handle device during PIN entry. Included list with supported cipher suites for TLS	Gustavo Cotta	Luis Gomes	12/2015
1.1	OpenSSL version updated to 1.0.2d.	Gustavo Cotta	Gabriel Rozenwald	09/2015
1.0	Initial revision	Gustavo Cotta	Diogo Lisita	06/2015

Additional Notes

Rev.	Observation
1.3	Alterations from previous version on page 14
1.2	Alterations from previous version on pages 8, 12 and 13
1.0	Initial version (derived from the former Security Policy)

Contents

List of Tables

List of Figures

Glossary

3DES	Triple DES
AES	Advanced Encryption Standard
API	Application Programming Interface
ARP	Address Resolution Protocol
BIOS	Basic Input Output System
DDR2 SDRAM	Double Data Rate Synchronous Dynamic Random Access Memory
DUKPT	Derived Unique Key Per Transaction
ECDHE	Elliptic Curve Digital Diffie-Hellman Exchange
ECDSA	Elliptic Curve Digital Signature Algorithm
ELF	Executable and Linking Format
GPRS	General Packet Radio Service
GSM	Global System for Mobile (communication)
HW	Hardware
ID	Identity
IP	Internet Protocol
MSR	Magnetic Stripe Reader
NFC	Near Field Communication
OS	Operating System
PA-DSS	Payment Application Data Security Standard
PCI	Payment Card Industry
PIN	Personal Identification Number
POI	Point Of Interaction
POS	Point Of Sale
PPP	Point-to-Point Protocol
PTS	PIN Transaction Security
RISC	Reduced Instruction Set Computing
RSA	Public key cryptosystem
SAM	Security Access Module
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
SW	Software
TCP	Transmission Control Protocol
TFT	Thin Film Transistor
TLS	Transport Layer Security
UDP	User Datagram Protocol
USB	Universal Serial Bus
VDC	Volts - Direct Current
WEP	Wired Equivalent Privacy
WAP	Wireless Application Protocol

Chapter 1

Introduction

This document aims to detail the proper use and the configuration of the Avixy 4000 terminal, in a secure manner, in order to meet the security requirements of the Payment Card Industry - PCI.

This document is focused on the terminal user and addresses the physical and operational characteristics of the terminal, the available services and functionalities and guidance on developing secure applications up to the decommissioning of the terminal.

Chapter 2

General description

2.1 Terminal overview

The Avixy 4000 (Figure 2.1) is a portable hand-held payment terminal, also known as a Point of Interaction (POI) device, able to perform both on-line and off-line secure electronic transactions.

Figure 2.1: Avixy 4000

It offers connectivity by GSM/GPRS Mobile Networks and features NFC/Contact-less, as well as smart card for chip-enabled card payment.

Additionally, it has a Magnetic Stripe Reader (MSR) and two Security Access Modules (SAMs), which can be used to enhance security and cryptography.

2.2 Transaction acceptance environment

The Avixy 4000 falls into the category of *attended POS devices* and is inserted in the subcategory of *standalone POS devices*, since it serves the single purpose of authorizing and clearing payment card transactions and requires the supervision of the vendor personnel during its operation - configuring its needs for an attended environment for operation.

2.3 Dimensions and weight

The Avixy 4000 is designed to be a comfortable hand-held device (and **shall** be used as this), enabling the user to hinder thirds from picking their PIN.

Operating the device in countertop or any other conditions, but hand-held, **will not** provide the required privacy for PIN entering.

Figure 2.2 illustrates the device portability, while Figure 2.3 depicts the device main dimensions. The device weighs about 500g.

Figure 2.2: Illustration of the device's portability

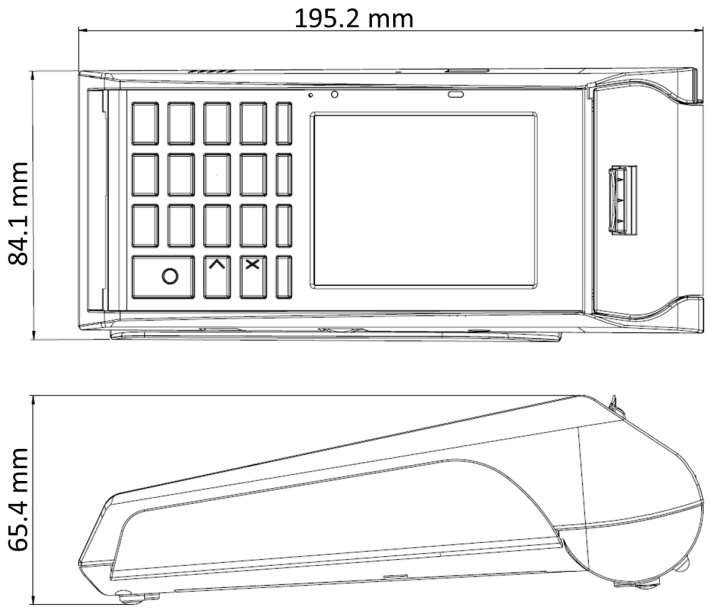


Figure 2.3: Avixy 4000 dimensions

2.4 Technical specifications

Figure 2.4 displays the main features of the terminal. Those specifications are further described in Table 2.1.

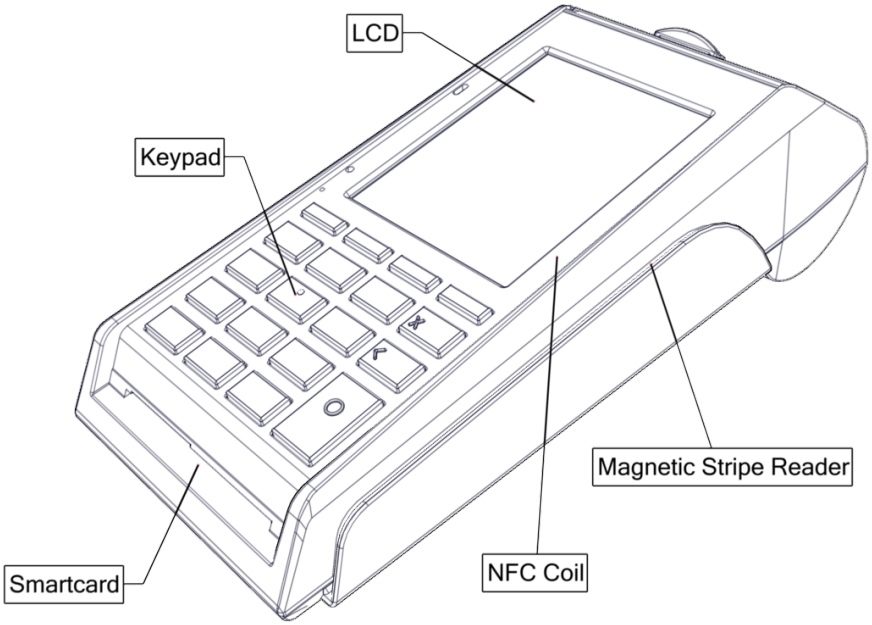


Figure 2.4: Avixy 4000 specification

Table 2.1: Main Specifications

Feature	Description
Security Processor	Secure Microcontroller, 32-Bit RISC
Application Processor	Secure Multimedia Processor, 32-Bit RISC
Main Memory	32MBytes, DDR2 SDRAM
System Flash-Memory	Micro SD 1GByte
Operation System	Custom Linux based on kernel *3.17*
Chip-based Card Interface	Smart card and NFC
Additional Credit Card Interface	Magnetic Stripe Reader
Connectivity	GSM/GPRS
Display	65K TFT Color Display, 240x320 pixels with touchscreen
Additional Peripherals	2xSAMs, Thermal printer mechanism, Speaker

2.5 Device identification

2.5.1 Terminal label

In order to identify the Avixy 4000 revision one shall refer to its label placed at the back casing of the device, see Figure 2.5. In Figure 2.5 the label displays: "Avixy 4000 3-2", where the Avixy 4000 device ID "3" and while the revision is an increasing number, in the example displayed is the revision "2" and "HW 2-2.0", where the Avixy 4000 hardware ID is "2" and the revision is "2.0".

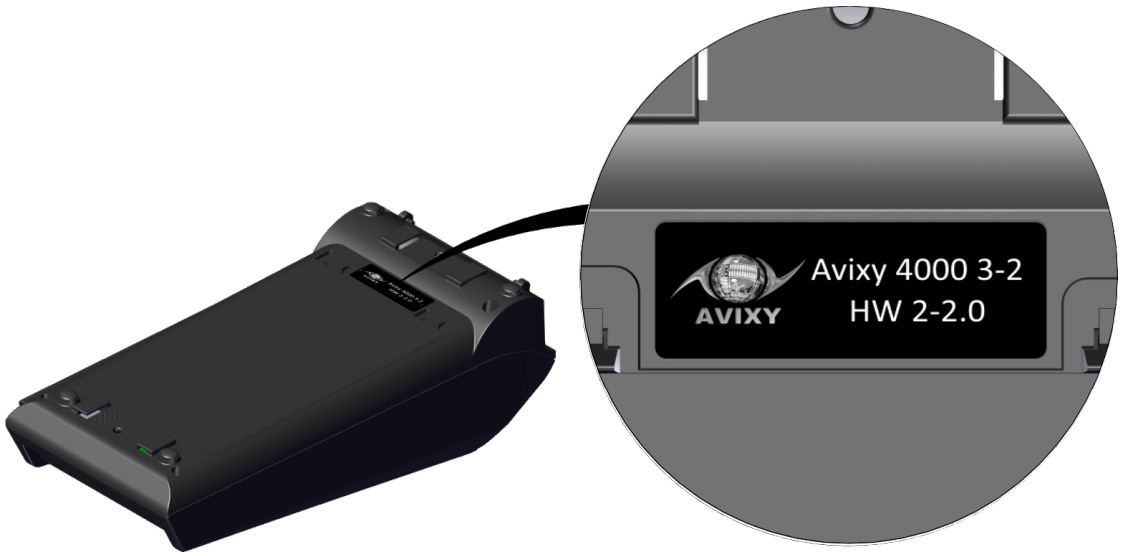


Figure 2.5: Avixy 4000 ID label

2.5.2 Terminal revisions

The product name, hardware (HW) and firmware (SW) versions are shown in the terminal display when the Operating System (OS) is booting-up (Figure 2.6). Additional information regarding firmware submodules can be retrieved by the administrator menu (see [1] for more details).

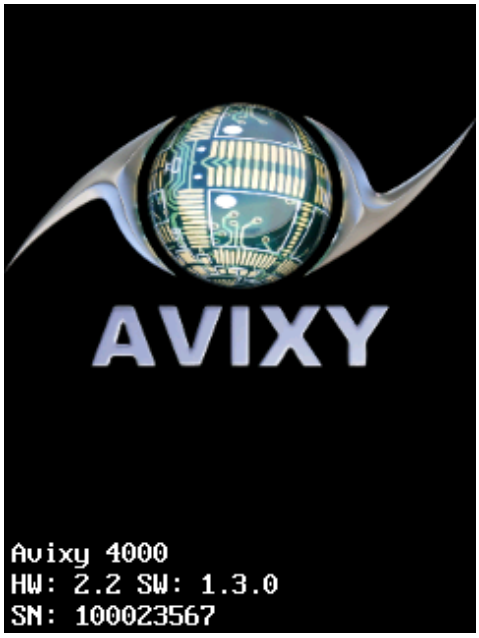


Figure 2.6: Avixy 4000 BIOS
Product name, hardware, software and serial number.

PCI approved hardware and firmware versions for the device can be checked at the PCI *Approved PIN Transaction Security (PTS) Devices* page:

`https://www.pcisecuritystandards.org/approved_companies_providers/approved_pin_transaction_security.php`

Chapter 3

Guidance

3.1 Visual inspection

To assess the authenticity of any Avixy 4000 terminal, one can refer to any of the identification and serial numbers on the device and contact the vendor directly.

The terminal **shall** be visually inspected daily, looking for any indication of tampering of the POS case and for any strange device plugged to the Point of sale (POS).

In case of any suspicion, violation of the security seal, or evidence of any shim device, do not use the terminal and contact the support immediately.

3.1.1 Security inspection

When receiving a new terminal via shipping, it **shall** be inspected for any suspicious modification or sign of tampering. Hereafter is provided some guidance on what to verify on both initial security check (when receiving terminal via shipping) and during the daily inspection:

- Look for any marks all over external surface;
- Check for any missing or altered screws;
- Check the integrity of the protection seal (*Section 3.1.2 Protection seal*);
- Look for any suspect surface;
- The appearance of the POS must be as the figures of this manual;
- Look for any differences from keypad described in this manual and the POS;
- Inspect the smart card slot (*Section 3.1.3 Smart card slot*);
- Inspect the MSR slot/rail (*Section 3.1.4 MSR slot (rail)*).

Mind that this list is not meant to be exhaustive, instead, it is meant to serve as a guideline on what can be checked.

3.1.2 Protection seal

There is a protection seal, which aids on identifying whether POS was opened (disassembled) or not. It covers one of the screws that keeps the terminal assembled.

The seal is located under the battery (see figure Figure 3.1 for reference).

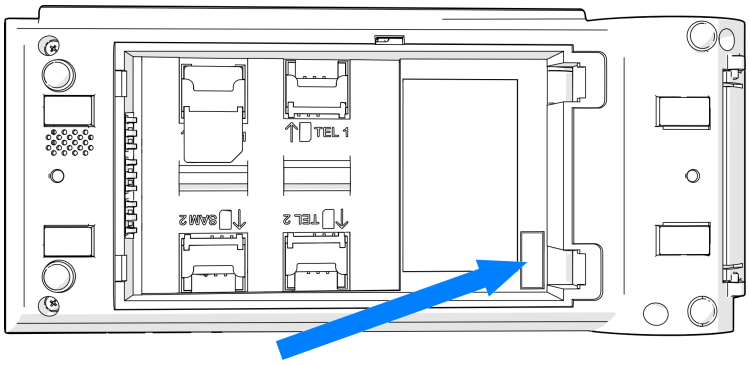


Figure 3.1: Illustration of the Avixy 4000 Protection Seal

3.1.3 Smart card slot

The smart card slot must be visually inspected daily. Look for any shim device/wires that may have been introduced in the slot. No loose component should be seen within it.

3.1.4 MSR slot (rail)

Similarly to the smart card slot, the MSR slot/rail shall also be periodically inspected. There must be no extra head or wires visible within it.

3.2 Device installation

The Avixy 4000 User Manual, [1], describes the basic operation instructions. Since it is a portable hand-held device no further configuration/setup is required.

3.2.1 Configuration settings

There is no sensitive configuration setting needed. No sensitive default value is necessary to be set prior use. Nevertheless, an application programming interface (API) is provided to the application developers so that they can securely develop applications to the device.

3.3 Personal data privacy

Each time the user types their PIN number sequence anyone who observes this, e.g. by looking over the user's shoulder, can memorize it and together with stolen or skimmed materials the user might be subject to criminal actions. Therefore, the user **shall** prevent it by making use of the fact that the Avixy

4000 is a handheld device and, therefore, they **shall** place the device close to their body during the PIN entry to avoid unauthorized observation of the entered digits. Mind that the user **shall** remove the device from any bedding the device is placed on top of prior to entering their PIN, Figure 3.2 illustrates the proper manner the user SHALL enter their PIN. This protection of the PIN can be extended to protect any user data.

(a) Improper way to hold the device during PIN entry (b) Proper way to hold the device during PIN entry

Figure 3.2: PIN entry protection

3.4 Maintenance

The terminal does not require periodic maintenance. In case of malfunction, the device should not be opened by non-authorized/certified personnel, it will trigger the tamper mechanisms that erase the stored keys and render the device useless for payment transactions until new keys are provided. Besides, maintenance should only be performed by the vendor or certified partners.

3.5 Decommissioning

Each country has its specific regulations on how to proper dispose of electronic devices. Therefore, regardless of national legislation, it is a liability of the vendor to guarantee that prior to disposal of the device - by any means - the device **shall** be electronically tampered - what can be done, preferably, by removing the rear casing.

3.6 Software development

Avixy follows a set of best practices to prevent any issues and bugs during the development process. Therefore, Avixy encourages the adoption of such practices during developing applications (software) for the terminal as well. A guidance of such practices is provided in [2]. Mind that some of those practices are not only nice to have, but mandatory in order to remain compliant with PCI PA DSS.

3.7 Patches and updates

The Avixy 4000 supports firmware and software updates. In any case, the update is done by the firmware who checks for the update authenticity prior to executing the updated firmware/application. Whenever this authenticity checks fails the update is rejected.

For the sake of the device security, Avixy recommends that always the latest version of the firmware should be used, unless otherwise stated.

Note:

The procedure regarding patches and updates, from the device perspective, is identical. The difference between the two of them lays on the development process considering the patches (bug fixing) and constant development for improving the device performance and security.

Chapter 4

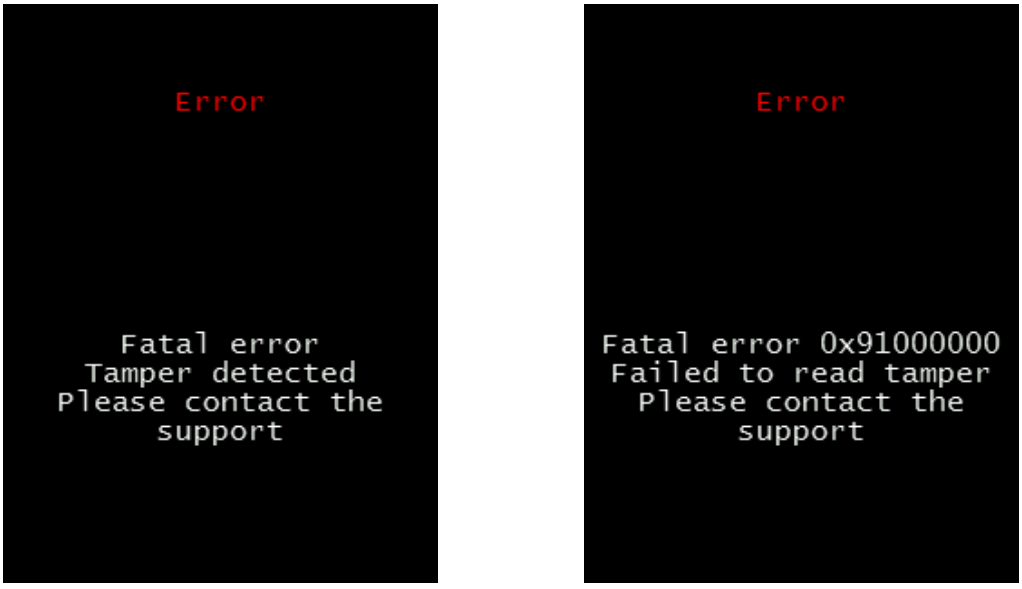
Security infrastructure

4.1 Tamper event

In the case of a tamper event, the sensitive keys stored on the device are promptly destroyed - by erasing the key encrypting key used to cypher the keys stored within the device. Additionally, if the machine is powered on, it is forced to reboot.

During the next boot, if the tamper condition has been re-established the device shows the screen on Figure 4.1a and the device is responsive to enter on the administration menu. If the tamper condition persists the device is completely inoperable and the keypad is also non-responsive, therefore, even accessing the administration menu is not possible. In this latter case the device displays the screen on Figure 4.1b.

Mind that in any scenario the application is not able to run again and the device **shall** return to manufacturing site for inspection and re-factoring - so that all protections are enabled and the keys are reloaded.



(a) (b)

Figure 4.1: Display content in event of tamper. (a) After tamper condition has been restored. (b) With tamper condition still active.

4.2 Self-test

The device performs signature checks for every portion of firmware on start-up. After that, the device's integrity is constantly monitored by the tamper detection mechanisms.

Firmware is only loaded into the memory after signature check, the same is valid for the drivers, managers and the application. Additionally, the firmware cannot be modified by the application at any time since the application runs on the user space of the kernel and have no privilege to change any memory portion if not explicitly allowed.

Nevertheless, a reboot is forced at least once every 24 hours - if in continuous operation - in order to perform the signature check of the firmware all over again. In the event of a failure during the self-test the following scenario is possible:

- BIOS verification failed: The bootloader points the boot to wait data from USB. Only signed and valid BIOS will be executed after loaded
- OS firmware verification failed: The machine resets - in case of a failure on the Linux portion of the firmware - or the machine presents a fatal error message
- Application verification failed: Invalid signature warning is displayed

In the event of tampering the device instantly destroys all sensitive data stored in the device and the device becomes inoperable, see *Section 4.1 Tamper event*.

4.3 Environmental and operational conditions

The Avixy 4000 operational and storage environmental conditions are presented in the Table 4.1:

Table 4.1: Environmental Conditions

Feature	Description
Operational Temperature Range	From 5 °C to 45 °C
Storage Temperature Range	From -20 °C to 55 °C
Allowed humidity conditions	85% @ 40 °C
Power Supply	5 VDC

Additionally, the environmental failure-protection mechanisms implemented within the secure processor will trigger in the following conditions:

Table 4.2: Voltage sensor thresholds

	V_{DD}	REG18V	V_{BAT}
Overvoltage threshold	3.85V	2.25V	3.8V
Low voltage threshold	-	-	2.5V

Exposing the device to any condition outside the operational conditions showed above will cause the device to trigger tamper mechanisms that makes the device inoperable.

Table 4.3: Temperature sensor thresholds

	Core temperature
High temperature threshold	+125 °C
Low temperature threshold	-45 °C

4.4 Communication protocols

The PCI PTS approval of the Avixy 4000 is only valid if the application uses only the open protocols in Table 4.4, which list the open protocols made available in the device.

Table 4.4: Communication protocols implemented in the Avixy 4000

Protocol Name	Component	Source Code Base and Version	Financial
Ethernet	Application Processor	Linux Kernel *3.17*	yes
WiFi	Application Processor	Linux Kernel *3.17*	yes
GPRS	GPRS Modem	Modem Vendor	yes
ARP	Application Processor	Linux Kernel *3.17*	yes
PPP	Application Processor	Linux Kernel *3.17*	yes
IP (General)	Application Processor	Linux Kernel *3.17*	yes
TCP	Application Processor	Linux Kernel *3.17*	yes
UDP	Application Processor	Linux Kernel *3.17*	yes
TLS	Application Processor	*OpenSSL 1.0.2d*	yes

4.4.1 Security protocols (TLS)

The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols as well as a full-strength general purpose cryptography library [3].

The PCI organization requires that the OpenSSL library must be part of the firmware and the application **shall** use the firmware implementation, instead of its own, in order to remain compliant with the PCI requirements. Therefore, in the Avixy 4000 firmware, the OpenSSL is part of the firmware and Avixy is committed to constantly update the firmware in order to provide a safe version of the OpenSSL library.

In order to guarantee safety in the OpenSSL usage, the following *MUST* be followed. Mind that non-compliance with the hereafter mentioned settings implies in non-compliance with PCI standards:

- use TLS v1.2 or higher;
- verify the authenticity of the trusted roots before passing them to OpenSSL;
- check the server certificate validity:
  - hash (>= SHA256);
  - key size (>= 2048 bits);
  - validity date (past or future);
  - trust chain.

- deny the following:
  - Self-signed certificates;
  - Not yet valid certificates (with future dates).

The following cipher suites are supported:

- RSA with 3DES EDE CBC SHA
- RSA with AES128 CBC SHA
- RSA with AES128 GCM SHA256
- RSA with AES256 CBC SHA
- RSA with AES256 GCM SHA384
- ECDHE RSA with 3DES EDE CBC SHA
- ECDHE RSA with AES128 CBC SHA
- ECDHE RSA with AES128 CBC SHA256
- ECHDE RSA with AES128 GCM SHA256
- ECDHE ECDSA with 3DES EDE CBC SHA
- ECDHE ECDSA with AES128 CBC SHA
- ECDHE ECDSA with AES128 CBC SHA256
- ECDHE ECDSA with AES128 GCM SHA256
- ECDHE ECDSA with AES256 GCM SHA384

Additionally, in order to assure a secure use of the OpenSSL library, the Avixy 4000 does not support SSL v2 and v3 protocols. Those protocols were removed from the compilation options of the OpenSSL library:

Listing 4.1: OpenSSL library configuration option

```
1 ./Configure linux-armv4 shared no-bf no-cast no-dh no-dsa no-md2 ↙
    ↳ no-mdc2 no-rc2 no-rc4 no-rc5 no-ssl2 no-ssl3 zlib
```

The application must implement session management and:

- keep track of all connections and restrict the number of sessions that can remain active to the minimum necessary number.
- set time limits for sessions and ensure that sessions are not left open for longer than necessary.

4.4.2 WiFi

Avixy 4000 supports only two encryption methods for WiFi[a]:

- WPA
- WPA2

Even though the communication done using OpenSSL hinders the issues provoked by the known WEP vulnerabilities, this encryption method is not supported. Therefore, one can use only WPA and WPA2 - where WPA2 **shall** be the default option, since it provides the strongest protection for secure-related transactions. Additionally, WPS is disabled on the Avixy 4000 firmware, eliminating the known vulnerability of WPA and WPA2.

4.5 Minimal configuration

The Linux kernel is configured with the least drivers necessary and contains only the software (components and services) needed.

The Linux operating system segregates virtual memory into kernel space and user space. Kernel space is strictly reserved for running the kernel, kernel extensions, and most device drivers. In this memory space, the processor has access to the hardware and has full control of the memory. In contrast, user space is the memory area where all user mode applications work. Each user space process normally runs in its own virtual memory space, and, unless explicitly requested, cannot access the memory of other processes.

This is the basis for memory protection, and a building block for privilege separation - a technique in which a program is divided into parts which are limited to the specific privileges they require in order to perform a specific task. This is used to reduce the potential damage of a security attack.

The application is also signed with the acquirer's key, which gives access to fewer Linux functions.

4.6 Application signing tool

Only authorized software is allowed to run in the Avixy 4000. Therefore, in order to enable application to run in the device, Avixy provides the application developers with a tool that implements a scheme allowing acquirers to securely sign their applications. This scheme implements both split knowledge and dual control techniques. Details of this implementation is given in [2].

4.7 Roles and services

No sensitive service or roles are used within the device. Sensitive data are handled by the firmware in a secure fashion using cryptographic authentication.

[a] The support to WEP has been removed from Avixy 4000 kernel.

Chapter 5

Key management

5.1 Design and techniques

The Avixy 4000 has a complex software design aiming to be secure. Its firmware uses internally the most established cryptographic algorithms:

- AES (128-256)
- RSA (1024-2048)
- 3DES (112-192)
- ECDSA (256)

Every driver, application, application certificate, application update, OS and BIOS are signed with single-purpose unique keys. All signatures are verified upon every system start-up. This signature verification mechanism guarantees that only reliable and trustworthy code is executed in the device.

The use of the POI with a key-management system, for encrypting PIN blocks on on-line transactions, other than Derived Unique Key Per Transaction (DUKPT) will invalidate any PCI approval of the POI. In *Section 5.2 List of keys* is presented a list of the application visible keys and their algorithms.

5.2 List of keys

The following list presents the Avixy 4000 Encryption and Signature keys and their main characteristics and responsibilities[a]

- $KTamper$**:**

  This AES-256 key is stored at POI Secure Master Key Storage and it is used to encrypt sensitive keys stored in the POI;

- $KAcquirer$**:**

  Generated by Acquirer (RSA2048);

[a] Some keys were omitted form this list because they are not visible to the application.

  - $KAcquirer_{PRIV}$:
    Stored at Acquirer's HSM;
    Signs the Acquirer's application on every version release.
  - $KAcquirer_{PUB}$:
    Stored at POI EEPROM;
    Verifies the signature of the acquirer's application on every POI boot.

- $KUserCertificate$:

  Generated by Avixy's HSM (RSA2048);

  - $KUserCertificate_{PRIV}$:
    Stored in Avixy's HSM;
    Signs the application key ($KAcquirer_{PUB}$) as soon as it is received from the Acquirer.
  - $KUserCertificate_{PUB}$:
    Hard-coded in the POI Operational System;
    Verifies the signature of $KAcquirer_{PUB}$, improving the acquirer's application verification.

- $BDKAcquirer_{Avixy}$:

  Generated by Acquirer (3DES 112);

  Stored on Avixy's HSM;

  Used to derive an unique IPK per POS;

  Used only for initial key loading.

- $BDKAcquirer$:

  Generated by Acquirer (3DES 112);

  Stored on Acquirer's HSM;

  Used to derive an unique IPK per POS;

- IPK (Initial PIN Encryption Key):

  Generated by Avixy's HSM (3DES 112);

  This key is derivated from $BDKAcquirer$ or $BDKAcquirer_{Avixy}$;

  After loaded into the POI, this key will generate the first set of the $DUKPTSet$ keys and will be erased immediately;

- $DUKPTSet$:

  Generated by POS (3DES 112);

  Stored on POI secure processor;

  Encrypts sensitive data of the transactions (e.g.: PIN).

  According to ANSI x9.24, this set of keys are unique per terminal.

- TK:

  Generated by POS (3DES 112);

  This key is not stored, it is erased just after use;

  Used to encrypt IPK;

  Has a mechanism to avoid replay attacks;

This key will be used when the Acquirer's HSM decide sending a new IPK to the POS. In this case, TK is generated by the POS and shared with HSM to start a secure communication channel to receive the new IPK (more details in *Section 5.3 Key injection*).

- $KEncTKAvixy$**:**

  Generated by Avixy's HSM (RSA 2048);

  - $KEncTKAvixy_{PUB}$**:**
    Hard-coded in the Avixy's firmware;
    Encrypts TK;
  - $KEncTKAvixy_{PRIV}$**:**
    Stored at Avixy's HSM;
    Decrypts TK;
    Used in initial key loading only.

- $KEncTK$**:**

  Generated by Acquirer's HSM (RSA 2048);

  - $KEncTK_{PUB}$**:**
    Hard-coded in the Avixy's firmware and in the Acquirer's application;
    Encrypts TK;
  - $KEncTK_{PRIV}$**:**
    Stored at Acquirer's HSM;
    Decrypts TK;

- SK**:**

  Session key (AES128 or AES256) for remote communication between the Server and the POI. This key is generated provided by the implementation of the TLS protocol under the OpenSSL library. This is a session key and lasts only during the open session, being erased with the session end.

The device may support the encipherment of the PIN multiple times as part of a transaction series; however, the PIN shall only be enciphered using the same PIN-encipherment key and transaction data, and not different keys or transaction data.

The application shall not use cryptographic tools to encrypt/decrypt arbitrary data. Each key has its function and cannot be misused. E.g.: Arbitrary data can't be encrypted using account data-encrypting keys.

Remainders:

- PIN-encryption keys must be used only used to encrypt PIN data.
- Key-encrypting keys must be used only used to encrypt keys.
- PIN keys shall never be used to encrypt keys.
- Key-encrypting keys shall never be used to encrypt PIN data.

5.3 Key injection

Key loading is not a sensitive service, since the sensitive data (transport key - TK - and the initial pin encrypting key - IPK) are not exposed, by any means, to any entity other than the secure cryptographic unit that securely stores the sensitive data. Nevertheless, the application developer is liable to guarantee that the private pair ($KEncTK_{PRIV}$) of the public key ($KEncTK_{PUB}$) used to cypher the transport key is indeed securely stored and handled by the acquirer's infrastructure.

5.4 Key replacement

Whenever the compromise of the original key is known, or even suspected, the referred key **shall** be properly replaced. Additionally, as defined in [4], whenever the time deemed feasible to determine the key by exhaustive attack elapses such keys **shall** be properly replaced.

REFERENCES

[1] AVIXY TECNOLOGIA. *Avixy 4000 - User Manual*. Rev. 1.6. [S.l.], January 2016.

[2] AVIXY TECNOLOGIA. *Application Development Guidance*. Rev. 1.3. [S.l.], January 2016.

[3] THE OPENSSL TEAM. *The OpenSSL Project*. `http://www.openssl.org/`. Accessed 16-April-2013.

[4] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, NIST. *Recommendation for Key Management - Part 1: General*. 2012. `http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf`. Accessed 25-May-2015.