Payment Card Industry (PCI)

PIN Transaction Security (PTS) Point of Interaction (POI)

Modular Derived Test Requirements

Version 6.2

January 2023

Document Changes

Date	Version	Description
February 2010	3.x	Initial Request-for-Comment Version
April 2010	3.0	Initial public release
October 2011	3.1	Clarifications and errata, updates for non-PIN POIs
February 2013	4.x	RFC version and combining of DTRs and DTPs
June 2013	4.0	Public release
June 2015	4.1	Updates for errata and new Core Section J. Added Device Management.
July 2015	4.1a	Minor errata
September 2015	4.1b	Section J updates
June 2016	5.x	RFC version
September 2016	5.0	Public release
March 2018	5.1	Modified B4, B9, B12, D1, D4, K1, K4, K12, and K16, and added K24 for SCRP approval class. Modified A1, A4, A5, A8, B2, F1, K1.1, and K11.2. Reorganized B20 and added reference to example of Security Policy layout in Appendix H. Modified side-channel testing appendix. Errata.
June 2020	6.0	▪ Restructured Modules. ▪ Added appendices for Domain-Based Asset Flow Analysis and Evaluation Guidance for CPUs. ▪ Modification of Visual Observation Deterrents criteria. ▪ Eliminated Removal Detection requirements. ▪ Added required support for ECC for POI devices that accept IC cards. ▪ Split requirement A1 into two separate requirements: 1) Tamper-Detection Mechanisms 2) Protection of Sensitive Keypad Inputs ▪ Split requirement A6 into two separate requirements: 1) Invasive Attacks for Cryptographic Keys 2) Non-invasive Attacks for Cryptographic Keys. ▪ Allow the inclusion of MSRs in SCRPs for use in SPoC solutions. ▪ Errata
March 2022	6.1	Added requirement for unauthenticated wireless communications.
January 2023	6.2	Modifications in support of mobile standards, ANSI reference changes, made references to PIN CVM Application more generic & B20

Contents

Introduction

General

The Test Requirements in this document were derived from the PCI PTS POI Security Requirements as embodied in the *PCI PTS POI Security Requirements*. These Derived Test Requirements (DTRs) are grouped into four major sections within this document:

- **DTR Module 1: Physical and Logical Requirements**
- **DTR Module 2: POS Terminal Integration Requirements**
- **DTR Module 3: Communications and Interfaces**
- **DTR Module 4: Life Cycle Security Requirements**

Structure of the DTRs

Each PCI requirement as stated in the *PCI PTS POI Security Requirements* is represented by a subsection. For example, Requirement A1 is represented in this document as:

DTR A1 Tamper-Detection Mechanisms

Each PCI requirement has been divided into component parts. These parts are identified by the corresponding PCI requirement number and a number distinguishing it from other components of the same requirement. These components parts are DTRs.

These are identified by a “T,” followed by the component identification number

For example, the first DTR under A1 is:

TA1.1

Reporting Requirements for PTS Laboratories

To be acceptable for reviewing, evaluation reports must present evidence of a device’s compliance to the Security Requirements. Before releasing a new test report, a delta report, or any updated report version, the lab must perform a thorough technical and quality assurance review to ensure that:

- The report accurately provides information specified in this document, without ambiguous or inconsistent information.
- The report and device details are complete and accurate.
- The report Includes information relevant to any applicable FAQs.
- The report conforms to Laboratory General Requirements and any other related documentation in the PTS Program, such as (but not restricted to) Reporting Guidance and Templates.

Minimum Contents of Reports and Minimum Test Activities

All reports shall include a device summary section at the beginning of the document. This summary shall include the following:

- A device overview that summarizes the device's design, hardware and software architectures, functionalities, and any other security-relevant attributes, features, or functions including (but not restricted to):
  - *Security processor(s) and other processors and memory, operating system(s), boot-up sequences, firmware modules, software applications, crypto functions, data-loading mechanisms, hardware versions and software versions with explanations of versioning, features and functions associated with the device's approval, etc.*
  - *This overview must present all security-relevant features necessary to derive assets, threats, and attacks relevant to the Security Requirements, as follows:*
    - *Photographs of the device from different perspectives, sufficient for all sides of the evaluated device to be clearly seen*
    - *A list or table of the device's features and functions*
    - *A list or table of the device's security-related assets relevant to applicable DTRs*
    - *A list or table of the device's principal physical components along with a photograph of the components, disassembled, indicating each of the components*
    - *Illustrations/descriptions of the device's logical architecture, interfaces, and key management, to indicate the hierarchy/relationships of firmware and other logical attributes of the device*
    - *Block diagram(s) indicating the hardware evaluation boundary with regard to the device's overall architecture and peripheral interfaces*
    - *If applicable, a diagram or description the device's integration into other architectures and/or integrating components*
    - *Clear definitions of which hardware and firmware features of the overall POI solution are within or without the scope of evaluation*
- A summary list of DTRs with verdicts on whether tested and whether compliant; and
- A summary of any assistance provided by the vendor to the lab.

In support of some test steps, as directed by the test laboratory, the vendor must support the laboratory in various tasks (such as, but not restricted to, code review, fuzzing interfacing, DPA, etc.) to avoid prohibitively lengthy test activities.

The vendor shall make all source code pertinent to Security Requirements available to the lab and provide assistance to make a systematic review of relevant security functions.

Evidence-based reporting, demonstrating device compliance through robust testing, is the fundamental basis for achieving device approval. For all DTRs, the tester shall state the following in writing:

- At the beginning of the report, a list of all documentation, including DTR sections and Technical FAQs version used during the evaluation

- Throughout the report, inline references to specific documents when addressing other sub-requirements
- For each DTR, the DTR definition and guidance (if any), followed by the sequence of all DTR work items—e.g., TA1.1, TA1.2, TA1.3, … etc.—directly preceding the report's explanations on each item. Cited DTRs text shall match the current DTRs version and shall be clearly distinguishable from Laboratory-generated report text. 'N/A' verdicts shall be clearly identifiable as such.

The evaluation report document shall demonstrate compliance to Security Requirements and shall present all test evidence as requested within individual DTRs. For all DTRs, the tester shall present sufficient information on direct tests and theoretical claims to validate conclusions by demonstrating how any conclusions are derived. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid against PTS Program—i.e., considering DTRs, FAQs, Program Guide, and any other related documents. Every DTR should be supported by sufficient evidence for the evaluation conclusions placed in the report to be understood and confirmed. This includes (but is not limited to):

- References to relevant information in the overview sections of the evaluation report, and to other DTRs where appropriate;
- Descriptions of the vendor's attestations of compliance to security requirements, with:
  - *Descriptions of information and assistance provided by the vendor to support the evaluation;*
  - *Accurate descriptions of relevant device attributes, for example (but not restricted to): physical and logical protections, chip architecture, OS, etc.;*
  - *Detailed explanations of the scope and focus of test activities and attack hypotheses, including explanations of white-box or black-box approaches used, and why;*
  - *Details of decisions made for performing penetration testing, the methodologies used, and the results of penetration testing;*
- Proofs of the reliability of testing reused between devices having similar characteristics;
- Justifications for any reliance on test evidence not derived directly from the evaluation activities;
- Justifications for any reliance on test evidence derived from devices other than the device under test;
- Explanations for any conclusions based on theoretical analysis instead of applied testing.

The tester shall detail where costing information is based on testing or assumptions and provide justification for any use of assumptions rather than actual testing.

The tester shall justify any deviation from the prescribed routines. The tester is not limited to only presenting information specified by DTR text/guidance/FAQs/Program Guide. Where necessary to support a conclusion, expand upon that information.

In most cases the DTR text is insufficient without combining photographs and/or other graphic illustrations that explain the evaluation. Images shall be of sufficient quality for relevant details to be viewed—for example, clear identification of a hardware component, relevant information clearly discernible in a graph, images capturing displays and other outputs, source code fragments, etc.

All DTRs must include references to documents and any other relevant sources of information upon which the evaluation relies. References must indicate information sources sufficiently to enable PCI to identify test evidence following device approval.

Where test evidence from prior evaluations is being reused, the similarities and differences between the prior and current devices must be detailed. Evidence of how the similarities were confirmed must also be presented.

Clear indications to the reader must be provided in any DTR where results have been reproduced from a prior report.

Re-use of test results for PTS POI v6.x evaluations is only allowed where the evidence is no older than the last prior major version—i.e., v5.x or other v6.x reviews may supplement work done in the current review.

Asset Flow Analysis

Guidance

The purpose of the Asset Flow Analysis is to describe in block-diagram form how assets travel within the device (both logically and physically) and are protected as they are processed and manipulated by it. The Asset Flow Analysis does not have to be provided as a single flow diagram. It can be made up from several flow diagrams so long as it is clear how each flow diagram is interconnected and/or interrelated.

Within the Asset Flow Analysis, appropriate domains are assigned for each logical and physical component in the device including software modules, hardware components, and PCB tracks (which may be grouped if several tracks combine a bus). The tester then uses this Asset Flow Analysis to scope the device and apply the appropriate DTRs for the specific domain.

Requirement

The vendor shall provide an Asset Flow Analysis highlighting each physical and logical component in the device and indicating how each asset flows between each physical component, showing the logical modules used to protect it. The general idea is to indicate the status of an asset as it travels through the device—for example, whether the asset is clear text or encrypted at a point in the data flow. Any hardware component or software module interfacing with the asset will be virtually marked (or tagged) with the domain that the asset belongs to as defined in Appendix G, "Domain-Based Asset Flow Analysis."

The Asset Flow Analysis shall be included in the PCI POI report and referenced in the appropriate DTR evaluations throughout. It is therefore expected that the vendor will perform this domain-based Asset Flow Analysis and provide the results and a complete explanation to the testers. The test lab will verify the analysis and use the effective domain rating as direction for the device evaluation.

It is important that asset flows and domains are correct and complete for each asset. The lab will validate and notify the vendor if discrepancies are discovered in the asset flows for corrective action.

DTR Module 1: Physical and Logical Requirements

A – Physical Security Derived Test Requirements

DTR A1 Tamper-Detection Mechanisms

The device uses tamper-detection and response mechanisms that cause it to become immediately inoperable and result in the automatic and immediate erasure of any sensitive data that may be stored in the device, such that it becomes infeasible to recover the sensitive data. These mechanisms protect against physical penetration of the device by means of (but not limited to) drills, lasers, chemical solvents, opening covers, splitting the casing (seams), and using ventilation openings.

Guidance

The objective of this section is to assess the device's ability to protect clear-text PINs and other sensitive data. Attack scenarios are not presented in this requirement. DTRs A2, A4, A6, A8, A10, and A13 include attack costings which will incorporate tamper-detection results from this requirement during the attack development.

Requirement A6 focuses on determination of secret or private keys. This requirement focuses on tamper-detection and response mechanisms in place to prevent disclosure of sensitive data.

"Immediate" is defined as fast enough to ensure erasure occurs before the tamper-detection mechanisms can be disabled.

For those devices that do not contain secret information, device disablement may be used in lieu of "immediate erasure of all secret information."

"Secret information" consists of any private or secret cryptographic keys or passwords/authentication codes relied on by the device to maintain security characteristics governed by PCI requirements.

The device is protected against penetration by including features that detect any feasible attempts to tamper with the device and cause immediate erasure of all cryptographic keys and sensitive data when such an attempt is detected.

Removal of the case or the opening, whether authorized or unauthorized, of any access entry to the device's internal components causes the automatic and immediate erasure of the cryptographic keys stored within the device.

Any tamper-detection/key-erasure mechanisms function even in the absence of applied power.

If any of these keys are not zeroized, then other mechanisms must exist to disable the device, and these keys must be protected in accordance with Requirement A6.

Secret or private cryptographic keys that are never used to encrypt or decrypt data (e.g., keys, accounts, PINs), or are not used for authentication, do not need to be considered secret data and therefore do not need to be erased—for example, where the device uses a chip set that automatically generates keys at initialization, but the keys are not subsequently used by the device.

(continued)

Acceptable epoxy for encapsulation possesses the following characteristics:

- *Opaqueness: Epoxy must be opaque in the visible spectrum.*
- *Hardness: Epoxy must be hard enough so that a sharp object cannot be used to penetrate the epoxy to the depth of the underlying circuitry.*
- *Tamper Evidence: The epoxy must show visible evidence of tamper when an attempt to penetrate the epoxy with a sharp object is made.*
- *Adhesion: Epoxy must resist attempts to forcibly separate it from the circuit board. When enough force is applied to remove the epoxy, severe damage should result such that the device is non-functional.*

If switches are used as the primary protection for the area around a physical keypad area, then at least three blind, tamper switches must be implemented. The switches must be protected from attacks that use the application of adhesives or conductive liquids to disable the switches. The design must ensure that a minimum of three switches in the keypad area must be individually attacked to disable them.

If tamper grids are used as a primary mechanism, they meet the following:

- *Use a minimum of two layers of internal grids for protection.*
- *Vias of "upper grid" must be protected separately to vias of "lower grid" (for example, the two tamper grids must not be connected by vias that are accessible on both grid layers, or vias must be protected by other tamper mechanisms, such as switches).*
- *Maximum width/separation (of active traces) of 6 mil.*
- *Use "opposing" tamper-responsive traces routed side-by-side on each layer.*

Demonstrating compliance to this requirement requires the inclusion of high-resolution images and/or diagrams indicating the principal security-relevant components of the device and the internal location of these components, such as (but not restricted to):

- *PCBs and FPCs*
- *Meshes*
- *Tamper circuit diagrams*
- *Keyboards, keys, and connecting components*
- *Screens and connecting components*
- *Connectors to card readers, boards, FPCs, screens, etc.*
- *Processors, microcontrollers, memory chips, etc.*
- *Tamper switches and components connecting to these*

SCRPs must be capable of providing information to a query from an external source (such as a payment application on a mobile phone) to indicate if in a tamper state.

Where applicable, testing shall be performed and validated with respect to Appendix H.

TA1.1 The tester shall examine the Asset Flow Analysis to verify that it is complete and accurate, and that it allows to unambiguously map all hardware components including PCB tracks, passive components, plastics, etc., to a security domain.

TA1.2 The tester shall provide an overview of the POI and how it is constructed. The tester shall include an exploded diagram of the POI showing how all sub-components are assembled and connected internally—for example, an explanation of processor (secure and unsecure) architectures and where these are located with regard to the internal areas of the device.

TA1.3 The tester shall describe any physical shielding—i.e., active, passive—or other chip or embedded physical protections that any microprocessors used in the device that contain clear-text secret or sensitive data, and how these are effective against tampering the physical package(s)—e.g., die encasement. The tester shall describe any physical barriers that surround microprocessors in the device, and state whether/how these totally envelop or partly surround microprocessors, and whether such barriers are tamper-responsive.

TA1.4 The tester shall enumerate each of the circuit boards indicated in the POI in the table below, providing, at a minimum:

a) The PCB designator (name) used;

b) The version of the PCB; the main purpose of the PCB;

c) Pictures of the front and back of each PCB and references thereto;

d) Note as to whether the PCB contains any sensitive signals (such as clear-text PINs, MSR, ICC, or display connections—but not tamper signals); and finally

e) Outline of the tamper-detection mechanisms used on that board (such as “4 tamper switches on lower face,” “6 tamper switches on upper face,” “two internal tamper grids,” etc.).

The tester shall adapt the table (for example, by adding columns or additional notes) as necessary, to present any additional information.

PCB Designator	PCB Version	PCB Purpose	Security Domain/Assets	Picture Reference	Sensitive Signals	Tamper-Detection Mechanisms

TA1.5 Using vendor documentation for each tamper grid that is implemented, the tester shall complete the details indicated in the table below, describing, at a minimum:

a) The location of the tamper grid (PCB designation and layer);

b) Its physical implementation (that is, its composition, for example copper on a rigid PCB, conductive ink on plastic, etc.);

c) The size of the conductive traces and the distance between each tamper-detecting trace (not necessarily between each trace) as well as the distance between layers for tamper grids which provide protection against penetration through the side of the PCB;

d) The method used to connect to the tamper grid (such as through hole via, buried via, soldered connection, elastomeric strip, etc.);

e) Whether traces from the same tamper signal are routed immediately adjacent to one-another without another tamper signal or passive signal interspacing them.

The tester shall adapt the table (for example, by adding columns or additional notes) as necessary, to present any additional information.

Tamper Grid Location	Physical Implementation	Size of Traces and Distance between Traces, Signals, or Layers	Number of Tamper-detecting Signals	Method of Connection	Adjacent Signals?

TA1.6 The tester shall describe what testing was performed to validate the protection provided by each of the tamper grids enumerated above.

TA1.7 For each tamper switch used in the POI, the tester shall complete the details indicated in the table below, at a minimum.

The tester shall use the ”Additional Comments” column to note any unusual features the tamper switch may possess that make it easier or harder to attack (such as being covered by a flexible tamper grid, or having a unique construction)

Switch Location	Number Used in that Location	Physical Implementation	Size of Switch Contacts	Conductive Ink Protections	Additional Comments

TA1.8 The tester shall describe what testing was performed to validate the protection provided by each of the tamper switches enumerated above.

TA1.9 The tester shall note which tamper-detection mechanisms use active high, active low, dynamic, resistive (or other) types of sensors. The tester shall confirm that any guard rings or interspaced traces in tamper grids are at opposing voltages that will activate tamper detection if electronically shorted. The tester shall note what testing has been performed to confirm this.

TA1.10 The tester shall describe any volume-encapsulation methods used in the device (for example, epoxy filling) that are designed to make penetration or reverse-engineering more difficult.

TA1.11 The tester shall describe what testing was performed to validate any volume encapsulation. Testing must include chemical and/or abrasive, and heating methods to bypass this protection.

TA1.12 The tester shall describe any attachment or "forming" methods (such as soldering, elastomeric strips, or adhesive for attachment, and plastic/metal walls for forming the shape of flexible circuits) used as part of the security features of the POI. For example, the tester shall detail the methods used to secure any flexible tamper grids, or "cover PCBs" so they cannot be bent or lifted out of the way.

TA1.13 The tester shall describe what testing was performed to validate any attachment and forming methods. Methods of testing must include use of localized heating, solvents, and abrasion. The tester shall justify why the testing performed was sufficient and why the security measures cannot be bent, melted, or otherwise bypassed, to gain access to sensitive signals.

TA1.14 The tester shall provide details on the security processor used in the POI, including how it drives tamper-detection features. The tester shall provide and reference a picture of the location and area surrounding the security processor.

TA1.15 The tester shall provide and make reference to a schematic diagram of the tamper circuits of the POI, showing connections to all tamper-detection features including switches and tamper grids.

TA1.16 The tester shall state how any passive components, connectors, or other items that carry tamper signals are protected against being accessed. The tester shall include any connections to power planes through hole vias that may be exposed outside of the tamper-detecting areas of the POI.

TA1.17 The tester shall describe how the POI responds to a tamper-detection event. The tester shall show the visible response(s) of the device's display (if any) upon tampering.

TA1.18 Deriving from previous descriptions, the tester shall explain how the immediate and complete erasure of all sensitive information from the POI results from tamper-detection events, and if applicable, where any of these keys are not zeroized, how other mechanisms exist to disable the device, and how these keys are protected in accordance with Requirement A6.

TA1.19 From the above descriptions the tester shall explain how the POI is rendered inoperative after any tamper event.

TA1.20 For SCRPs, the tester shall validate that the SCRP is capable of providing information to a query from an external source (such as a payment application on a mobile phone) to indicate if in a tamper state.

DTR A2 Protection of Sensitive Keypad Inputs

There is no demonstrable way to disable or defeat the tamper-mechanism/s and insert a sensitive key-press-disclosing bug.

Keypads used for PIN entry require an attack potential of at least 26 per device for identification and initial exploitation, with a minimum of 13 for initial exploitation, exclusive of the IC card reader, as defined in Appendix B.

Keypads used for manual PAN entry, but not PIN entry—e.g., a non-PED—require an attack potential of at least 16 per device for identification, with a minimum of 8 points for initial exploitation.

Guidance

Sensitive keypads are defined as those through which clear-text PINs or manual PANs can be entered.

Protection of manual PAN entry is only applicable if the device is being assessed against the SRED feature.

When designing an attack against the device, replacement of casing parts like the front and rear case of the device shall be considered as part of the overall attack.

A2 allows the evaluator to use any method of attack feasible against the terminal limited only by the attack potential of 26 in total with 13 for initial exploitation for PIN entry, and 16 in total with 8 for initial exploitation for account-data entry. The POI device must be able to withstand attack from any side, including front and rear case replacement up to the attack potential value.

The entire keyboard circuit must be evaluated.

For reference, see information gathered in prior DTR.

TA2.1 The tester shall describe how the sensitive keypad entry mechanism(s) are implemented in the POI.

TA2.2 The tester shall describe the path taken by the signals that connect the sensitive keypad entry mechanism(s) to the secure processor(s). The tester shall reference the relevant aspects of the asset flow.

TA2.3 The tester shall list any components—including passive parts or segments, connectors, or other items—that are connected to the path of the customer PIN or manual PAN signals. The tester shall reference the relevant aspects of the asset flow.

TA2.4 For each PCB that carries sensitive keypad signals, the tester shall describe what tamper-detection mechanisms protect these signals from being accessed (such as tamper grids). The tester shall confirm that these mechanisms protect the entire path taken by the signals, as described above.

TA2.5 The tester shall describe whether any of the items on the path of the keypad signals are not protected by a tamper-detection mechanism. For example, the tester shall note if a signal via terminates on the same layer as a tamper grid and whether any passive components are located outside of the protected area or are connected to vias (including power vias) that terminate outside of the protected area(s).

TA2.6 The tester shall describe any POI security features used to protect sensitive keypad data that has not already been covered in the previous descriptions (for example, special processor packaging). The tester shall detail what testing has been performed to validate each of these features.

TA2.7 The tester shall explain how the device is designed to mitigate against overlays.

TA2.8 The tester shall explain how the POI is protected against an internal overlay being placed across the keypad.

TA2.9 The tester shall explain how the POI is protected against keypad attacks from all sides of the POI, including the front and rear of the device.

TA2.10 The tester shall describe the different attack paths considered. Using the format shown in Appendix B and providing images allowing the principal steps in the analysis to be understood, the tester shall generate sensitive keypad attack calculations using different attack techniques on the POI, and present the most feasible attacks for capturing PIN, and if SRED, the most feasible attack for capturing PAN data—i.e., presenting at least one distinct example of each. The attacks should be dissimilar in approach unless the lab can fully justify the infeasibility of any second divergent approach. The tester shall state explicitly where testing has verified any specific stage of the attack—including the time, equipment, and skill required, and number of mechanisms to bypass—and where assumptions are used in place of testing. The tester shall justify why any assumptions have been used instead of than actual testing. Calculations shall include evidence justifying particular rating levels as being appropriate.

See the "Attack Examples" section of Appendix B for detailed examples of appropriate presentation of attack calculations.

DTR A3 Robustness Under Changing Environmental and Operational Conditions

The security of the device is not compromised by altering:

- ***Environmental conditions***
- ***Operational conditions***

(An example includes subjecting the device to temperatures or operating voltages outside the stated operating ranges.)

Guidance

The requirement focuses on the robustness of the tamper-detection and tamper-response functionality under changing environmental and operational conditions. Attack paths to manipulate such tamper-detection and response mechanism are the focus of Requirement A1; attempts to determine keys, for instance by fault-injection techniques, are covered by requirement A6.

The vendor must either provide substantive data to support the security of the product functioning outside normal operating conditions or show that the product uses sensors that will trigger a tamper response.

The objective is not to replicate the vendor testing, but instead it is to account for shortcomings within the vendor's testing of the implementation.

The tester may rely upon vendor testing as appropriate to fulfill the following test steps.

TA3.1 Using the schematics and descriptions from TA1.2 through TA1.9, the tester shall accurately list the temperature and voltage ranges for all components included in the tamper-detection and response circuit. This shall include mechanical switches and active elements (but not passive elements such as resistors and capacitors).

TA3.2 The tester shall use the table below to accurately detail the environmental protection features implemented by the POI. For voltage, the tester shall outline which voltage is being monitored (for example, external, main processor core voltage, battery voltage, etc.). If more than one voltage monitoring is provided, the tester shall detail all of these. For each environmental factor monitored, the tester shall detail what circuitry is performing this monitoring and what occurs when during an out-of-range detection.

	Maximum Value	**Minimum Value**	**Detecting Circuitry**	**Response**
Voltage (Specify type)	Configured Value	Configured Value		
	Tested Value	Tested Value		
Temperature	Configured Value	Configured Value		
	Tested Value	Tested Value		

TA3.3 In the maximum/minimum values for each item, the tester shall note what the vendor attests the value is set to in the “Configured Value” cells of the above table.

TA3.4 Using a POI that has been configured by the vendor (using special test code from the vendor, which shall be removed from production units) to operate self-tests such that the correct operation of the device can be confirmed. The tester shall test each of the environmental features listed above and enter the value at which the detection circuitry activates into the “Tested Value” cells of the above table; or the vendor should provide sufficient test reports covering all required tests.

TA3.5 The tester shall detail whether the self-test program used above executed correctly at all times during each of the tests above, within the ranges before activation of the environmental detection circuitry.

TA3.6 Given the details and results above, the tester shall justify why the tamper-detection and response mechanisms will remain functional and the POI secure at all extremes within the range of environmental monitoring.

DTR A4 Protection of Sensitive Functions or Information

Sensitive functions or data are only used in the protected area(s) of the device. Sensitive data and functions dealing with sensitive data are protected from unauthorized modification without requiring an attack potential of at least 26 for identification and initial exploitation, with a minimum of 13 for initial exploitation, exclusive of the IC card reader, for identification and initial exploitation, as defined in Appendix B.

Guidance

Public keys used for functions that impact security requirements, such as firmware updates, display prompt control, or remote key distribution schemes must be protected against modification and substitution. Secret and private keys used for functions that impact security requirements must be protected against modification, substitution or disclosure.

Protected area of the device is that area(s) within the boundaries of the tamper-detection and response mechanisms.

The lab shall consider glitch attacks including (but not restricted to): voltage and EM glitching. At a minimum, these should consider the core and battery input for the security processor.

Where applicable, testing shall be performed and validated with respect to Appendix H.

TA4.1 The tester shall verify the completeness of the information regarding sensitive information and functions presented by the vendor.

TA4.2 In the following table, the tester shall outline the locations of all types of sensitive information and functions, adding to those provided where other types of sensitive information exist within the POI. This shall include both long-term and temporary storage locations, as well as information on any programmable logic used in the POI as part of the PIN storage/processing/entry circuit. The storage area column shall outline where and what type of storage is used for this information (such as internal SRAM of the security processor, or external Flash memory); and the Method of Protection column should outline what mechanisms of the POI design protect this information (such as encryption, signature(s), physical protection, etc.). The method of protection description should be explicit about this protection, detailing specifically which cryptographic keys are used for encryption, for example, or which physical protection mechanisms are utilized.

It is expected that each type of information may have multiple “storage areas” and “methods of protection” (for example, clear-text PINs may appear in internal memory of the security processor, as well as on the external heap cache and within the processor registers, and the external memory may be protected by encryption with the external bus key as well as with physical protections of the secure area of the POI).

The tester shall adapt the table (for example, by adding columns or additional notes) as necessary, to present any additional information.

Security Domain/Asset (Sensitive Information)	Storage area	Method of protection
PIN/Clear-text PINs		
PIN/Passwords/authentication codes		
PIN/POI Firmware		
Key/Public keys		

TA4.3 The tester shall verify that clear-text public keys only exist within a certificate or a secure cryptographic device and that it is not possible to illegitimately substitute one key for another.

Public keys not stored in certificates or in a secure cryptographic device must be stored encrypted or have a MAC (message authentication code) created using the algorithm defined in *ISO 16609*, in order to ensure authenticity and integrity.

TA4.4 Using the information from the table above, the tester shall provide details on the different integrated circuits (memory, processors, programmable logic etc.) that are used to store, process, or secure sensitive information.

TA4.5 For each of the integrated circuit elements (described above) which may be programmed or configured in some way, the tester shall enumerate:

a) The different ways in which that element may be programmed or configured (for example, JTAG).

b) Any in-circuit testing or debugging features provided by these elements.

c) Unused features and programming that should be disabled.

TA4.6 The tester shall detail what methods have been implemented to disable all of the programming/testing features outlined above. The tester shall detail the testing performed to validate that these features are indeed disabled. The tester shall justify why these measures are sufficient and confirm that these features cannot be re-enabled.

TA4.7 If additional memory is implemented and is not included in the sensitive-information storage areas above, the tester shall detail what processes have been used to validate that this is the case. The tester shall detail all memory in the device and detail where sensitive data is stored and how it is protected.

TA4.8 If the POI allows for execution of applications and firmware on the same processor that stores or operates on clear-text passwords/authentication codes, PINs, or public keys, the tester shall note what mechanisms are implemented to prevent these applications from modifying this information. The tester shall detail how this has been validated as sufficient.

TA4.9 Where signatures are used as a method of protection, the tester shall:

a) Validate that only approved algorithms and key lengths are used for the signatures.

b) Detail what padding scheme is used for the signatures and justify how this prevents attacks such as padding oracle attacks.

c) Detail when the signatures are validated, and how modification of the sensitive information is prevented after signature validation.

TA4.10 Where encryption is used as a method of protection of sensitive information, the tester shall:

a) Confirm that the encryption uses approved algorithms and key lengths.

b) Note what mode of operation is used for the encryption.

c) Justify how this prevents the re-location of memory from one area to another.

d) Justify how the method of encryption prevents the exposure of sensitive information through building of a “dictionary” of possible encrypted values.

e) If a key stream mode of encryption is used—for example, OFB—justify how the encryption of different data with the same key is prevented.

TA4.11 Where physical protections are used as a method of protection (for example, when clear-text information is stored in external memory), the tester shall:

a) Confirm that the physical protections cover all memory traces, vias, passive elements, or other areas of access. For example, detail whether the vias of the memory bus appear on the same layer as one of the tamper grids.

b) Detail how the memory packages are protected, including access to BGA balls and traces on internal chip carriers of packages.

TA4.12 The tester shall analyze the device's susceptibility to glitch attacks, including, but not restricted to, voltage and EM glitching. At a minimum, the tester shall consider the core and battery input for the security processor. Where applicable, the tester shall also consider embedded memory (SRAM, EEPROM, Flash, and ROM).

TA4.13 The tester shall describe and produce a costing for the most feasible attack to recover sensitive information from the POI. The tester shall detail for each step whether the information is based on testing or assumptions and provide justification for any use of assumptions rather than actual testing.

If an attack scenario can be developed that yields an attack potential of less than 26 per device for identification and initial exploitation or less than 13 per device for initial exploitation only, as defined in Appendix B, the vendor assertion cannot be verified. If only attack scenarios can be developed yielding attack potentials above these thresholds, the tester shall present these to demonstrate how the device is compliant to this DTR. At least one attack scenario shall be presented, in a format consistent with the examples shown in Appendix B in this document. Calculations shall include evidence justifying particular rating levels as being appropriate.

The tester may perform any test needed to validate the attack scenario.

The tester shall present evidence of the test methodologies followed and the validation results.

DTR A5 Monitoring During PIN Entry

There is no feasible way to determine any entered and internally transmitted PIN digit by monitoring sound, electro-magnetic emissions, power consumption or any other external characteristic available for monitoring—even with the cooperation of the device operator or sales clerk—without requiring an attack potential of at least 26 for identification and initial exploitation with a minimum of 13 for initial exploitation, as defined in Appendix B.

Guidance

For A5, monitoring sound refers to other audible sounds apart from the beep generated by the device when a key is pressed.

Methods such as video monitoring and shoulder surfing are addressed in A9.

TA5.1 The tester shall provide a circuit diagram of the input power circuitry of the POI, including any elements used to provide isolation of the power and EM emissions of the device.

TA5.2 Using an in-line resistor, current probe, or other suitable method, the tester shall monitor the (external) current drawn by the POI when pressing each of the ten numeric buttons during a PIN entry function. The tester shall ensure methods are implemented to trigger the captures at the same time (for example, use the signal that drives the sounding device of the POI). The tester shall detail the method used, providing photographs of the test setup.

TA5.3 The tester shall analyze the power captures obtained, as stated above, in both the time and frequency domains to determine whether any of the button presses provide a unique pattern that may be used to distinguish that button press from all others. The tester shall detail analysis and justify any conclusions drawn, referencing images where suitable as evidence. Generally, an initial observation of obtained signals is insufficient to validate that sensitive information does not leak. It is necessary to perform signal analysis to validate that an attacker may not straightforwardly discern sensitive data-dependency from power captures. If discernible, the DTR is not met.

TA5.4 Examine the layout and signal structure of the PIN input component (keypad) and determine how any sensitive information leakage by EM emissions may occur. Summarize findings.

TA5.5 Using suitable probes or antennas, the tester shall monitor the external EM emissions of the POI when pressing each of the ten numeric buttons during a PIN entry function. The tester shall ensure methods are implemented to trigger the captures at the same time (for example, use the signal that drives the sounding device of the POI). The tester shall detail the method used to capture the emissions, providing photographs of the test setup and justifying the reason why any particular location of the POI was used as the point to capture the EM emissions (in preference to any other location).

TA5.6 The tester shall analyze the EM emissions (EME) obtained above in both the time and frequency domains to determine whether any of the button presses provides a unique pattern that may be used to distinguish that button press from all others. The tester shall detail analysis and justify any conclusions drawn, referencing pictures where suitable as evidence. Generally, an initial observation of obtained signals is insufficient to validate that sensitive information does not leak. It is necessary to perform signal analysis to validate than an attacker may not straightforwardly discern sensitive data-dependency from EM emissions.

TA5.7 Using suitable microphones, the tester shall use the microphones to monitor the acoustic signals of the POI when pressing each of the ten numeric buttons during a PIN entry function. The tester shall detail the methods used to capture sounds, providing photographs of the test setup and justifying the reason any particular location of the microphones was used (in preference to any other locations).

TA5.8 The tester shall analyze the acoustic captures obtained in TA5.7 in both the time and frequency domains to determine whether any of the button presses provides a unique pattern that may be used to distinguish that button press from all others. The tester shall detail analysis and justify any conclusions drawn, referencing pictures where suitable as evidence. Generally, an initial observation of obtained signals is insufficient to validate that sensitive information does not leak. It is necessary to perform signal analysis to validate than an attacker may not straightforwardly discern sensitive data-dependency from acoustic signals.

TA5.9 The tester shall develop attack scenarios to defeat or circumvent the protection mechanisms against the monitoring of sound, electro-magnetic emissions, power consumption or any other external characteristic available for monitoring, using attack scenarios. If an attack scenario can be developed that yields an attack potential of less than 26 per device for identification and initial exploitation or less than 13 per device for initial exploitation only, as defined in Appendix B, the vendor assertion cannot be verified. At least one attack scenario shall be presented, in a format consistent with the examples shown in Appendix B in this document.

Monitoring must be done outside the protected areas of the device components (such as the PIN entry device's tamper-protected casing, or ICC reader) and must investigate any data emanating from inside these device components.

DTR A6 Invasive Attacks for Cryptographic Keys

Determination of any PIN-security-related secret or private cryptographic keys resident in the device by penetration of the device requires an attack potential of at least 35 for identification and initial exploitation with a minimum of 15 for initial exploitation, as defined in Appendix B.
Determination of any account-data-security-related secret or private cryptographic keys resident in the device by penetration of the device requires an attack potential of at least 26 points for identification and initial exploitation with a minimum of 13 for initial exploitation, as defined in Appendix B.

Guidance

Keys resident in the device or its components means clear-text secret or private keys.
If the encrypted keys are protected in accordance with the minimum key sizes and parameters for the key-encipherment algorithm(s) used as stipulated in Appendix E, they do not need to be considered.

Secret or private cryptographic keys that are never used to encrypt or decrypt data—e.g., keys, accounts, PINs—or are not used for authentication, do not need to be considered secret data and therefore are not subject to this requirement.

All physical and logical protections shall be evaluated and reported. Only protections that can be demonstrated as both enabled and efficacious shall be used to determine attack potential.

Where applicable, testing shall be performed and validated with respect to Appendix H.

TA6.1 The tester shall provide details on the type, location, and accessibility of the security-relevant processor(s) used by the POI, and any other elements of the POI that have relevance to possible attacks. This shall include a list of all barriers obstructing access to keys, beginning with the device exterior case and ending with inner-most barriers, clearly indicating which barriers are tamper-responsive, or not. The tester shall reference information previously supplied in DTRs A1 and A4 where applicable. The tester shall reference the relevant aspects of the asset flow.

TA6.2 The tester shall provide details on how cryptographic keys are stored and managed within the POI. The tester shall reference this information to the table provided in DTR A4, allowing the location of all applicable keys to be defined. The tester shall detail the testing performed to confirm the storage locations listed are correct. The tester shall reference the relevant aspects of the asset flow.

TA6.3 The tester shall provide details on any specific protections provided by the security processor (or other processors if applicable) that are designed to obstruct obtaining or determining the values of cryptographic keys. If specific protections are unknown, the tester shall provide details on protections strongly inferred through testing.

TA6.4 The tester shall:

a) Describe what protections the cryptographic processing elements implement to protect against glitch attacks to force cryptographic errors.

b) Refer to testing and results from DTR A4 where applicable.

c) Review the source code of the POI to confirm that any protection measures relied upon are enabled.

d) Describe why any secure boot-up operations are implemented appropriately to obstruct glitch attacks, referring to any available literature and vulnerability disclosures to support this or otherwise perform practical penetration tests to demonstrate this.

e) Describe why invasive and/or glitch attacks intending to change flags—e.g., flipping a bit value enabling a defense—are effectively obstructed.

TA6.5 The tester shall describe what protections are implemented within the cryptographic processing elements to protect against physical attacks at the chip level to extract the cryptographic keys. The tester shall review the source code of the POI to confirm that any protection measures relied upon are enabled and effective. The tester shall also determine whether protections can be disabled and how.

TA6.6 Referring to the information provided in DTR A4, the tester shall perform a review of available literature and vulnerability disclosures to confirm that the programming or in-circuit testing features of the processing elements of the POI cannot be re-enabled (either temporarily or permanently). The tester shall validate all documentation provided by the vendor.

TA6.7 If the POI stores clear-text cryptographic keys within external memory, the tester shall detail the physical security methods implemented to protect this memory. Note that PCB-based tamper grids are not considered sufficient to protect clear-text cryptographic keys.

TA6.8 The tester shall describe and cost the most feasible attack to recover cryptographic keys from the POI, using the above information. The tester shall detail whether steps are based on actual testing or on assumptions and provide justification for any use of assumptions rather than actual testing. This information should include, at minimum:

- The steps needed in any penetration.
- Attacks involving some or all of attacks modeled elsewhere. For example, DTRs A1, A2, A4, and A7 (but not restricted thereto) must be considered and included in this attack, if relevant.

The tester is not required to perform the attack entirely but may perform all or part of the attack to verify its validity. The calculation shall be based on the methodology depicted in Appendix B.

- If an attack scenario (including any method related to A1, A2, A4, and A7) can be developed that yields an attack potential for any PIN security-related cryptographic key of less than 35 per device for identification and initial exploitation or less than 15 per device for initial exploitation only, as defined in Appendix B, the vendor assertion cannot be verified. At least one attack scenario shall be presented, in a format consistent with the examples shown in Appendix B in this document. Calculations shall include evidence justifying particular rating levels as being appropriate.
- If an attack scenario can be developed that yields an attack potential for any account-data-security-related key of less than 26 per device for identification and initial exploitation or less than 13 per device for initial exploitation only, as defined in Appendix B, the vendor assertion cannot be verified. At least one attack scenario shall be presented, in a format consistent with the examples shown in Appendix B in this document. Calculations shall include evidence justifying particular rating levels as being appropriate.

The tester shall present evidence of the test methodologies followed and the validation results.

TA6.9 If the attack costing for DTR A4 was found to be less than the minimum points required for this DTR, the tester shall justify why the attack for DTR A4 cannot be used to recover cryptographic keys.

DTR A7 Non-invasive Attacks for Cryptographic Keys

Emanations from the device (including power fluctuations) cannot be feasibly used to recover PIN and/or SRED security-related cryptographic keys resident in the device.

Guidance

The vendor shall provide mechanisms to facilitate side-channel testing. These mechanisms shall include at least the following: an interface, the ability to vary data and keys, and the ability to set trigger points (for testing purposes only and not for production units), allowing the evaluator to perform analysis with direct access to security-relevant processor(s).

Where a device exclusively supports DUKPT or similar unique-key-per-transaction methodology for the protection of sensitive data, such as PINs, static-key statistical side-channel analysis using large batches of recordings does not need to be applied for these keys.

Secret or private cryptographic keys that are never used to encrypt or decrypt data—e.g., keys, accounts, PINs—or are not used for authentication, do not need to be considered secret data and therefore are not subject to this requirement.

SCA tests shall be performed in accordance with Appendix F including the scope of side-channel testing necessary to validate the device's compliance based on the identification of relevant keys below and taking into consideration the appropriateness of testing re-use and demonstrably effective countermeasures.

Notwithstanding physical protections (which may be validated by other DTRs), approved devices must not contain cryptographic implementations which, under analysis, can be straightforwardly compromised through SCA.

To pass this requirement, it must be demonstrated that any static value keys resist SCA attacks by expert-level skill using state-of-the-art tools analysing at least 100,000 demonstrably high-quality recordings, in-line with the details of this DTR.

***For PIN security-related keys,** the evaluation should determine at least one algorithm to analyze thoroughly, applying the SCA approach most likely to succeed, based on a rigorous assessment of asset value versus feasible attacks. Reasoning for not testing any algorithm must be explained. This reasoning should include reliable assumptions made in the vulnerability analysis based on asset value and attack complexity—for example, limits on collections such as delay insertions or key usage counters, and any additional countermeasures.*

***For SRED security-related keys,** the evaluation shall:*

- *Either reuse test evidence for PIN keys, if PIN key attacks have been selected for focus and/or if SRED-keys have the same implementation;*
- *Or perform SEMA or SPA only (see Appendix F) at a sufficient depth of investigation to show that cryptographic algorithms processing SRED keys have active and effective SCA countermeasures.*

(continued)

Even if physical protections against tampering are robust, and even in the presence of electronic noise and algorithmic complexity obscuring signals, it is expected that an attacker can defeat any undefended cryptographic implementation repeatedly exercising static keys. A device must have at least one demonstrably effective SCA countermeasure protecting all cryptographic algorithms in scope of this requirement. For software implementations and hardware implementations with known countermeasures, the evaluation shall identify one countermeasure, at least, and show its effectiveness. For hardware implementations with unknown countermeasures, the evaluation shall demonstrate the implementation's SCA characteristics are consistent with having at least one effective countermeasure.

Where applicable, testing shall be performed and validated with respect to Appendix H.

TA7.1 The tester shall identify and show in a table:

- A list of all PIN security-related cryptographic keys (secret and private) that may be directly attacked by side-channel analysis approaches. For SRED devices this includes account-data-security-related cryptographic keys. This list shall indicate whether the most applicable approach is statistical analysis of static keys with variable inputs or outputs, or other approaches such as, but (not restricted to) timing analysis.
- A list of all PIN security-related keys that may be indirectly attacked by side-channel analysis—for example, keys that may be attacked following additional attack steps. For SRED devices this includes account-data-security-related cryptographic keys.
- Any specific key-management techniques that either prevent or obstruct side-channel analysis, such as unique keys per operation or other constraints on exercising static key values or any other algorithmic constraints on SCA attacks.

TA7.2 The tester shall check the evidence provided by the vendor on the implemented cryptographic operations and detail all of the different cryptographic operations implemented within the POI. The tester shall verify whether they are implemented in software or hardware, and what side-channel analysis protections are implemented to protect each of these operations. The tester shall describe methods used for each side-channel protection (for example, time variance, masking, dual-rail logic, etc.). If it is not possible to determine this information from evidence provided by the vendor (for example if SoC vendor information is proprietary), then evaluation testing should be used to infer this information. Note that any operation that only functions using public keys does not require side-channel protections to be implemented.

TA7.3 Verify the items above, generally using source-code review, that the side-channel protection methods are implemented. For example, if the POI relies on protections provided by the processor hardware cryptographic engine, the tester shall confirm that the registers that enable this protection are correctly set by the POI firmware before every use of this cryptographic engine. If the protections are provided by the firmware, the tester shall check that the implementation is as described by the vendor. This evaluation activity may be focused at security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TA7.4 The tester shall perform preliminary side-channel analysis on the POI to characterize the cryptographic algorithms used to process sensitive data and/or operate with secret keys. Utilizing characterization results and knowledge of the POI physical design and software, the tester shall decide which side-channel attack paths provide the best opportunities for an attacker to compromise high-value assets. The tester shall develop side-channel analysis to explore methodologies most likely to achieve retrieval of a PIN security-related cryptographic key or keys derived from initial scoping. The tester shall develop these investigations to derive a demonstrable level of assurance consistent with any reported attack cost rating(s).

The tester shall describe any assistance provided by the vendor to ensure efficient side-channel testing—for example command scripts, event triggers, access to the processor, etc.

TA7.5 The tester shall justify the methodologies used and findings, in accordance with Appendix F and with regard to published attacks. The tester shall outline why analysis parameters provide a high level of confidence that key recovery through side-channel analysis is not feasible on this device.

Justifications of these to be reported shall include, but are not restricted to:

- Equipment used, and a summary of test parameters such as collection elapsed time and analysis elapsed time,
- Whether the attack can be feasible at any distance away from the processor,
- The number of sample recordings acquired,
- Sampling frequencies,
- Alignment methods,
- Signal analysis / filtering techniques,
- Correlation function(s) used,
- How optimum quality data collection was achieved, etc.
- How adequate key selection function coverage was applied.

Evidence to support this shall include (but is not restricted to) annotated graphical profiling of side-channel results such as:

- SPA/SEMA characterization,
- Noise-removal test results,
- Alignment test results,
- Demonstrations of non-sensitive data leakage,
- Correlation results for sensitive data leakage modeling,
- Timing analysis,
- DPA/DEMA attacks,
- Results validation and checks.

The tester shall present evidence of the effectiveness of active countermeasures implemented by the vendor in obstructing analysis.

Where no definite sensitive data leakage is achieved, validate why this is the case and provide explanations. For example (but not restricted to) by showing that I/O data leakage successfully localizes sensitive cryptographic operations, but key leakage is prevented by demonstrably effective countermeasures.

The evaluation may rely upon appropriate evidence available from existing side-channel evaluation testing to replace some of the testing workload described here. Such evidence must be no older than the last prior major version—i.e., v5.x or other v6.x reviews may supplement work done in the current review—of the PCI POI Security Requirements prior to the current evaluation's submission. If leveraging separate evidence, it is necessary to justify that this evidence is fully in scope of DTR A7 security requirements.

DTR A8 Physical Security of Display Prompts

The unauthorized alteration of prompts for non-PIN data entry into the PIN entry keypad such that PINs are compromised—i.e., by prompting for the PIN entry when the output is not encrypted—cannot occur without requiring an attack potential of at least 18 per device for identification and initial exploitation with a minimum of 9 for initial exploitation, as defined in Appendix B.

Guidance

A8 is applicable to a device that contains a display and may output non-PIN data.

A8 applies to any components or paths containing clear-text display signals between the cryptographic processor and display unit. B15 applies to devices that allow for updates of prompts or use cryptography to communicate with a display, whether performed by the vendor or the acquirer. C2.4 is appropriate for unattended devices that do not meet any of the aforementioned.

"Non-PIN data" refers to numeric data other than the PIN that is entered via the keypad. It does not include control inputs such as "enter," "cancel," etc. It also does not apply to data entered while the device is in special modes—e.g., maintenance—that are not intended to be accessed by cardholders and merchants.

Audio prompts must be considered if applicable.

TA8.1 The tester shall describe whether the POI allows for entry of non-PIN data to be passed external to the POI and whether that data is protected during the transfer. This non-PIN data must not be encrypted using the PIN key of the POI. The tester shall complete the following steps if the POI provides such functionality.

TA8.2 The tester shall describe the path from the display to the processing element that controls the display. Specifically, the tester shall note whether this is the security processor that handles PIN entry and/or cryptographic keys, or whether a different processing element is used. The tester shall include information on any connectors, cables, or other sub-components that lie in this path.

TA8.3 The tester shall describe the physical protections implemented to secure access to the display and path from the display to the controlling processing element. The tester shall make specific note of the following:

a) Whether the display sub-component can be removed from the POI without tamper detection; the tester shall detail any physical modifications that would be required to remove the display in this way (for example, cutting of POI casing around the display).

b) What protects the signals controlling the display on the module interface (cable, connector, etc.)? If physical protections are used, the tester shall detail at which point these protections terminate, and how this is sufficient to prevent access to signals on the module interface at all locations.

TA8.4 The tester shall detail where prompts used for non-PIN entry are stored within the POI, and describe the protections implemented to protect these prompts. The tester shall reference this information to the table of sensitive information provided in DTR A4.

TA8.5 The tester shall list any components—including passive parts or segments, connectors, or other items—that are connected to the path of the display signals.

TA8.6 The tester shall explain how the POI is protected against display attacks from each of all sides of the POI, including the front and rear of the device.

TA8.7 The tester shall describe whether any of the items on the path of the display signals are not protected by a tamper-detection mechanism.

TA8.8 The tester shall describe and provide a costing for the most feasible attack to change the prompts on the display of the POI. The tester shall detail where costing information is based on testing or assumptions and provide justification for any use of assumptions rather than actual testing.

If an attack scenario can be developed that requires an attack potential of less than 18 per device for identification and initial exploitation or less than 9 for initial exploitation per device as defined in Appendix B, the vendor assertion cannot be verified. At least one attack scenario shall be presented, in a format consistent with the examples shown in Appendix B in this document. Calculations shall include evidence justifying particular rating levels as being appropriate.

The tester may perform any test needed to validate the attack scenario.

The tester shall present evidence of test methodologies followed and validation results.

DTR A9 Visual Observation Deterrents

The device provides a means to deter the visual observation of PIN values as they are being entered by the cardholder.

Guidance

Some markets may require Option A.1.1 or A1.2 of Appendix A or a stricter criterion.

The vendor shall provide a privacy shield providing protections as described in Appendix A of this document. Alternatively, the vendor may use less restrictive privacy-shield criteria provided that the vendor supplies rules and guidance as to how the visual observation is to be deterred by the environment in which the device is installed. These rules shall be binding for the organization placing the device into the environment—for example, the acquirer or merchant. If the vendor gives rules for an external physical privacy shield, the vendor shall provide a demo/sample with the appropriate dimensions. The user (acquirer or merchant) instructions provided by the vendor shall clearly state that the acquirer or merchant must either meet the implementation criteria or deploy devices meeting the criteria defined in Appendix A, Section A1.1 or A1.2.

If the means to deter visual observation are not an integral part of the PIN entry device, the vendor shall specify by appropriate means (for example, drawings and description) how the visual observation is deterred by the structure or piece of equipment housing the device. These specifications shall be binding for the vendor.

TA9.1 The tester shall examine the means to deter the visual observation of PIN values provided by the device, and/or as described in the device documentation, to verify the assertions of the vendor. If the device provides physical observation deterrents, the tester shall show one or more images of these to support any conclusions.

TA9.2 The tester shall describe whether the POI requires external power or communications connections or is intended to be operated with no external cable connections.

TA9.3 The tester shall review any marketing literature and/or operations manuals provided with the POI, and state whether any of these explicitly describe handheld or desk-mounted operation or add-on components such as privacy shields. The tester shall detail any literature used to reach this conclusion.

TA9.4 The tester shall note whether the POI is intended to be deployed equipped with a privacy shield or whether the POI casing provides any fixture points, recess, or other indications that a privacy shield may be provided.

TA9.5 The tester shall note whether the POI provides any screw points or other fixtures designed to facilitate the mounting of the POI into a stand or other receptacle that would preclude the device’s being operated in a handheld mode.

TA9.6 When a device is claimed as handheld, the tester shall enter details of the POI into the table below. Accurate measurements should be taken with the POI configured so that it is at the maximum size and weight it would have during normal operation—for example any add-on modules, paper rolls, batteries, etc. should be installed before taking the measurements.

Dimension	Device Measurement	Maximum for Classification as Handheld
The width at the "5" key		7.62 cm
The height at the "5" key The sum of the width and the height at the "5" key		— 10.16 cm
The keypad length, from the bottom of the "0" key to the top of the "2" key		10.16 cm
The weight of the POI		500grams

***Note:** For a horizontal layout, exchange "width" and "length."*

TA9.7 Using the above information, the tester shall state whether the POI is designed such that handheld operation is enforced. The tester shall justify the conclusions.

TA9.8 If the device provides a privacy shield, the tester shall complete the table below with angles of observation to the center of the "5" key. If the observation angle is taken from an angle other than the absolute horizontal plane (not the "flat" plane of the POI casing) the tester shall justify why this is the case.

Angle of POI	Angle of observation to "5" key	Minimum angle required by Appendix A1.1	Minimum angle required by Appendix A1.2
0			
45			
90			
135			
180			
225			
270			
315			

The tester shall consider the examples included in Appendix A, Section A.2, of this document when evaluating the vendor's visual-observation deterrence rules. The user (acquirer or merchant) instructions provided by the vendor shall clearly state that the acquirer or merchant must either meet the implementation criteria or deploy devices meeting the criteria defined in Appendix A, Section A1.1 or A1.2.

TA9.9 If the means to deter visual observation are not an integral part of the PIN entry device, the vendor shall specify by appropriate means (for example, drawings and description) how the visual observation is deterred by the structure or piece of equipment housing the device. These specifications shall be binding for the vendor and specified in the vendor security policy described in B20. The tester shall examine this specification to deter the visual observation of PIN values provided by the device to verify the assertions of the vendor. The tester shall verify that the vendor-provided user guidance is clear, and that the measures provided can compensate for the lack of a privacy shield.

TA9.10 The tester shall present sufficient evidence and/or references for the above, for compliance to this DTR.

DTR A10 Magnetic-Stripe/Contactless Readers/Key Entered

The device protects all account data upon entry for magnetic-stripe or contactless data, and there is no method of accessing the clear-text account data to determine or modify the data (using methods described in A2) without defeating the security of the device. Defeating or circumventing the security mechanism requires an attack potential of at least 16 for identification and initial exploitation, with a minimum of 8 for initial exploitation.

Note: Contact chip is addressed in A13. Manual PAN entry is addressed in A2.

Guidance

Skimming is the unauthorized capture and transfer of payment data to another source, for fraudulent purposes.

Countermeasures include, for instance, active detection of skimmers, active disturbance of the skimming process. The protection of the reader may consist of resistance of the device cabinet/the reader enclosure against manipulation.

Skimming attacks to recover payment card data may occur via either the attachment of external devices or attacking other areas (hardware or software) of the device. Both must be considered for this requirement.

Access to the inside of the device for routine maintenance (for example, replenishing paper) shall not allow access to clear-text account data, for example, by making cabling which transmits the data physically inaccessible to routine maintenance personnel or encrypting the sensitive card data transmitted internally within the device between components.

Protections must exist against attacks wherein a bug is installed on the opposite side of the card track of the original MSR such that the attacker would only capture card data if the cardholder swipes the card with the track side facing the wrong way. This is due to some MSRs that are intentionally designed to capture the track data regardless of which way the card is swiped. Thus, cardholders become conditioned to swiping the card from either side, even where the reader does not support.

Demonstrating compliance to this requirement requires the inclusion of high-resolution images and/or diagrams showing the different readers and their locations.

As per the asset flow diagrams, all methods of card-data entry that are natively supported by the SRED firmware must be assessed. Magnetic stripe and contactless are addressed in this DTR. A device validated to SRED cannot have any card-reader types as part of the approved hardware and firmware version identifiers where that reader could not meet this requirement, A2 and/or A13, as applicable. Nor can the firmware support the receiving of card data from an external component that does not meet the same criteria.

The path for contactless data must be secured to 16 points from the point of digitization of this data. The point of digitization occurs when the data is processed by the NFC controller and not at the point of entry. The NFC controller acts as a modem converting the analog signal to a digital signal just as a magnetic-stripe reader or smart-card reader reads data and converts that to a digital signal. In all cases, the point of digitization is where the wireless signal is converted to a digital data stream.

All methods of access to the card data should be considered, including emanations (except for contactless).

SCRPs may include an MSR but cannot exclusively contain only an MSR. SCRPs must include contact and/or contactless chip card functionality. SCRPs must not contain a hybrid (combined contact chip and MSR) reader.

TA10.1 For SCRPs, the tester shall confirm that the SCRP does not exclusively contain an MSR, but also contains contact and/or contactless chip card functionality.

TA10.2 For SCRPs, the tester shall confirm that the SCRP does not contain a hybrid reader.

TA10.3 The tester shall describe whether the POI allows for capture of data from the magnetic stripe of a payment card. If the device processes magnetic-stripe information but does not integrate a magnetic-stripe card reader, the tester shall detail how magnetic-stripe information is obtained by the POI and provide any APIs used for this purpose. The tester shall only complete steps A10.3 through A10.9 if the POI has an integrated magnetic-stripe card reader.

TA10.4 The tester shall describe the location and operation of the magnetic-stripe card reader. The tester shall show one or more images of the device’s magnetic-stripe reader and associated hardware to support any conclusions.

TA10.5 The tester shall describe the path from the magnetic-stripe card reader to the security processor, including any cables, connectors, or other sub-elements on this path.

TA10.6 The tester shall describe whether the magnetic-stripe card reader implements logical protections (for example, encryption). The tester shall detail the following:

a) The integrated circuits used to provide the encryption, and any physical protections provided to these elements.

b) What algorithm, mode of operation, and key management are used for this purpose.

c) How cryptographic keys are loaded into the read head. The tester shall note whether keys can be updated after initial loading, and whether the POI supports this (for example, through an API or internal function of the security processor).

d) The method used to generate the keys loaded into the read head, and how this ensures that these keys are unique to each POI device.

TA10.7 If the device implements physical protections for the magnetic-stripe card reader, either in addition to or in lieu of logical protections, the tester shall detail the physical protections implemented to protect this path. The tester shall justify how this is sufficient to protect the entire path of the magnetic-stripe card signals from the read head to the security processor, including all vias, traces, connectors, and the pins on the read head itself.

TA10.8 The tester shall provide accurate measurements of any free space around the magnetic-stripe read head. The tester shall justify why placement of a secondary read head (even one that is very small and/or thin) is made infeasible by the POI design, either through lack of space and/or through the physical protections of the POI. The tester shall check for any free space on the opposite site of the magnetic-stripe read head to analyze whether a reader with the capability to read heads regardless of the way the card is swiped can be placed.

TA10.9 The tester shall identify whether the device implements any physical or logical ICCR/MSR combinations—for example, if the hybrid reader facilitates both skimming of the magnetic stripe and capture of the PIN during an ICCR “dip” read operation. In this case, the tester shall describe how there is no residual vulnerability to attacks on the combination reader intending to harvest both clear-text PINs and magnetic-stripe data.

TA10.10 The tester shall explain how the POI is protected against attacks for MSR data from all sides of the POI, including the front and rear of the device.

TA10.11 The tester shall describe the location and operation of the contactless card reader. The tester shall show one or more images of the device's contactless reader (antenna) and associated hardware to support any conclusions.

TA10.12 The tester shall list any components and describe the path from the point of digitalization to the secure processor—including passive parts or segments, connectors, or other items—that are connected to the path of the digitized contactless card data.

TA10.13 The tester shall explain how the POI is protected against attacks for contactless data from all sides of the POI, including the front and rear of the device.

TA10.14 The tester shall develop attack scenarios to penetrate the device to make any additions, substitutions, or modifications to either the device hardware or software, in order to determine or modify any account data, with an attack potential of at least 16 per device for identification and initial exploitation, with a minimum of 8 for initial exploitation. The tester shall detail where costing information is based on testing or assumptions and provide justification for any use of assumptions rather than actual testing.

If an attack scenario can be developed that requires an attack potential of less than 16 per device for identification and initial exploitation or less than 8 for initial exploitation per device as defined in Appendix B, the vendor assertion cannot be verified. At least one attack scenario shall be presented for each method of account-data capture supported—i.e., MSR and contactless—in a format consistent with the examples shown in Appendix B in this document

The tester may perform any test needed to validate the attack scenario. The tester shall determine the appropriate tests and whether the attack will be performed in its entirety or in part to verify the theory, and why.

The tester shall present evidence of the test methodologies followed and the validation results.

DTR A11 Account-data Processing

All account data is either encrypted immediately upon entry or entered in clear text into a secure device and processed within the secure controller of the device.

Guidance

The objective of this requirement is to ensure that all account data is handled in a secure manner. The requirement allows for the encryption of account data directly at the read head or for account data to be submitted to the device in clear-text form. This data is then communicated to a secure controller where it is processed.

The term "processed" is used as a generic term, which includes but is not limited to account-data encryption and the selective disclosure of clear-text account data by the secure controller to cryptographically authenticated applications (per B23.1).

For an authenticated application:

- *The application must reside and execute within the physically and logically secure boundary of the target of evaluation.*
- *The application must be cryptographically authenticated by keys secured within the key domain of the POI using algorithms and keys sizes consistent with those stipulated in B10.*

The minimum that must be encrypted, if present, under SRED, is:

- *Full track or equivalent (when aggregated as a single data element), including both Track 1 and Track 2;*
- *Manually entered security validation value—e.g., CVV2, CVC2, and CID2;*
- *Issuer discretionary data (as a single, unparsed field);*
- *Issuer discretionary data – sensitive data (as a parsed field). Any or all portions of the discretionary data are considered to be sensitive unless known to be non-sensitive;*
- *The PAN itself. If the PAN can be parsed, the parts of the PAN in clear text must not exceed the maximum truncation requirements of the associated payment brand when considered in totality with all possible firmware output methods.*

  For example, SRED-compliant POI devices are permitted to output clear-text PAN data to authenticated applications, as well as based on firmware authenticated whitelists. To support acceptable truncation formats for each Payment Brand, a POI may pass the clear-text data to an application for processing, or it may perform truncation and/or encryption of parts of the PAN based on firmware-authenticated whitelists. Whitelists allowing for digits of the PAN to be output in clear text when the system is operating in a format-preserving encryption mode must be considered in addition to any native truncation methods implemented by the POI.

In addition, the following must be encrypted if it is feasible to associate with the corresponding clear-text PAN:

- *Cardholder name*
- *Expiration date*
- *Service code*

In all cases cardholder name, expiration date, and service code must be encrypted for SCRs intended for use with COTS devices and for SCRPs.

(continued)

Clear-text account data cannot leave the secure boundary of the device except as part of a whitelist function. This applies whether or not it is encrypted at the point of capture (e.g., read head).

An SCR intended for use with a COTS device or an SCRP shall not release account data in the clear, even via a whitelist mechanism.

The POI shall only display one clear-text digit at a time during manual (key-entered) PAN entry. The display is required to obfuscate the digit prior to the display of the next clear-text digit. No more than one clear-text digit may be displayed at any time during entry.

TA11.1 The tester shall examine the device to verify that the asserted protections exist, how they are effective, and how they conform to the descriptions provided by the vendor in documentation. This will include disassembly of the test device when necessary.

TA11.2 The tester shall verify that the application handling clear-text account data executes within the secure boundary of the device. This should correlate with information presented in the Asset Flow Analysis or else constitutes an exception.

TA11.3 The tester shall validate that any manual PAN-entry functions implemented by the POI firmware never display more than one clear-text PAN digit at a time, and that any clear-text digit that is displayed is obfuscated prior to the display of the next digit.

DTR A12 Account-data Protection – Integration

The logical and physical integration of an approved secure card reader into a PIN entry POI terminal does not create new attack paths to the account data. The account data is protected from the input component to the secure controller of the device—i.e., it is not possible to insert a bug that would disclose sensitive data.

Guidance

The objective of this requirement is to assess those terminals where the card reader is integrated into the final solution and to ensure that as an integrated device it does not create any new weaknesses or permit new attack methods to be used against the data.

The ICC reader may consist of areas of different protection levels: the areas of the IC card itself, and the area holding retracted cards.

Note: *Secret or private cryptographic keys that are never used to encrypt or decrypt data, or are not used for authentication, do not need to be considered sensitive data and therefore do not need to be erased—for example, where the device uses a chip set that automatically generates keys at initialization, but the keys are not subsequently used by the device.*

Access to the inside of the device for routine maintenance (for example, replenishing paper) shall not allow access to clear-text account data—for example, by making cabling that transmits the data physically inaccessible to routine maintenance personnel or encrypting the sensitive card data transmitted internally within the device between components.

TA12.1 The tester shall verify that all testing procedures specified for Requirement A1 in the physical requirements have been satisfied in relation to the protection of account data.

TA12.2 The tester shall examine the device to verify that the asserted protections exist, how they are effective, and how they conform to the descriptions provided by the vendor in documentation. This will include disassembly of the test device when necessary.

DTR A13 ICCR Protection

It is neither feasible to penetrate the ICC reader to make any additions, substitutions, or modifications to either the ICC reader's hardware or software, in order to determine or modify any sensitive data, without requiring an attack potential of at least 20 for identification and initial exploitation, with a minimum of 10 for initial exploitation, as defined in Appendix B, nor is it possible for both an IC card and any other foreign object to reside within the card-insertion slot.

SCRPs shall require an attack potential of at least 26 for identification and initial exploitation, with a minimum of 13 for initial exploitation.

Guidance

The card reader may consist of areas of different protection levels, for example, the areas of the IC card interface itself, and the area holding retracted cards.

All contact points must be evaluated

The ICC reader may be equipped with mechanical and/or optical mechanisms to meet this requirement when used in conjunction with the implementation guidance.

Implementation guidance is provided to facilitate detection of shim devices by the entities (for example, merchants) deploying these devices.

A PIN-disclosing bug shall be prevented from being inserted into the device through the card slot. The volume of space accessible via the card slot that could be utilized by an attacker can vary with the geometry of the space and attack methods. For this reason, the requirement does not prohibit or specify a maximum volume. Rather, the feasibility of effective bug placement is to be considered when assessing A13 compliance. Examples of these considerations are:

- *Contact points must be present for the bug to connect to;*
- *The bug and wires must not obstruct normal operation;*
- *The placement of the bug must not cause tamper evidence that would be noticed by a typical cardholder.*

Space accessible via the IC card slot large enough to conceal a PIN-disclosing bug is not allowed. There must not be space accessible via the card slot large enough to conceal an IC card chip and small battery.

Demonstrating compliance to this requirement requires the inclusion of high-resolution images and/or diagrams showing the ICCR and its location.

DTRs A1.4, A1.5, and A1.7 must be completed for the ICCR where any specific references to "PIN" are to be read as "account data."

TA13.1 The tester shall examine the device to verify that the asserted protections exist and conform to the descriptions provided by the vendor in documentation. This will include disassembly of the test device when necessary.

TA13.2 The tester shall examine the implementation guidance to determine the adequacy of the documentation to facilitate the prevention and/or detection of the successful implanting through the ICC slot of a sensitive data-disclosing bug aiming at capturing offline PIN and IC card information.

TA13.3 The tester shall develop an attack scenario to enlarge the slot and describe this.

TA13.4 The tester shall perform a simulated transaction whilst inserting two unembossed cards into the slot. The device must not allow the successful execution of a transaction while two juxtaposed, un-personalized (un-embossed) cards are simultaneously inserted, each card with the minimum ISO 7810 thickness. And the IC card insertion slot height must be as small as possible along its full width. If it is possible to insert two cards and perform the transaction, the device does not comply with this requirement. The tester shall accurately measure the IC slot width, height (or different heights), and depth and record these, referring to an image of the IC slot opening location.

TA13.5 The tester shall provide and reference a picture of the area of the POI where the ICC acceptor is located.

TA13.6 The tester shall provide and reference a picture of the ICC acceptor itself. The tester shall note the manufacturer and model of the acceptor used.

TA13.7 The tester shall detail any active detection mechanisms the ICC acceptor utilizes to prevent a “shim” from being left in the slot. Where active protections are not present, the tester shall provide an extract from the merchant document that details how the slot can be checked for such items on a daily basis.

TA13.8 The tester shall provide and reference a picture of the internal space of the ICC acceptor (this may require the disassembly of one or more acceptors). The tester shall note whether there is any area within the acceptor that is larger than 10mm x 10mm in area, and has a depth in excess of 5mm, and any methods used by the vendor design to fill such areas.

TA13.9 The tester shall describe the path taken by the signals that connect the I/O pin of the customer ICC acceptor to the security processor.

TA13.10 As per the asset flow diagrams, **t**he tester shall list any components, including ICC interface components, passive components, connectors, or other items, that are connected to the path of the customer ICC I/O signal.

TA13.11 For each PCB that carries the customer ICC I/O signal, the tester shall describe what tamper-detection mechanisms protect these signals from being accessed (such as tamper grids). The tester shall confirm that these mechanisms protect the entire path taken by the I/O signal, as described above.

TA13.12 The tester shall describe whether any of the items on the path of the I/O signal are not protected by all tamper-detection mechanism. For example, note if a signal via terminates on the same layer as a tamper grid or if any passive components are located outside of the protected area or are connected to vias (including power vias) that terminate outside of the protected area(s).

TA13.13 Where not previously covered in Requirement A1, for each tamper grid that is relied upon for security of the customer ICC I/O signal, the tester shall complete the table below, including:

- Details on the location of the tamper grid (PCB designation and layer);
- The “physical implementation” (for example, whether it is composed of copper on a rigid PCB, conductive ink on plastic, etc.);
- The size of the conductive traces and distance between each tamper-detecting trace (not necessarily between each trace) as well as the distance between layers for tamper grids which provide protection against penetration through the side of the PCB; the method used to connect to the tamper grid (for example, through hole via, buried via, soldered connection, elastomeric strip, etc.); and finally

- Whether traces from the same tamper signal are routed immediately adjacent to one another without another tamper signal or passive signal interspacing them.

The tester shall adapt the table (for example, by adding columns or additional notes) as necessary, to present any additional information.

Tamper Grid Location	Physical Implementation	Size of Traces and Distance between Traces, Signals, or Layers	Number of Tamper-detecting Signals	Method of Connection	Adjacent Signals?

The tester shall describe what testing was performed to validate the protection provided by each of the tamper grids enumerated above.

TA13.14 From the above description the tester shall justify how the customer ICC I/O path is protected from interception at all points. Specifically, the tester shall note whether it is possible to intercept the signal with a probe, such as a metal pin (either straight or bent) inserted through a hole in the casing of the POI at any point.

TA13.15 From the above description the tester shall explain how the POI is rendered inoperative after a tamper event in the ICC-acceptor area.

TA13.16 The tester shall explain how the POI is protected against attacks for ICC data from all sides of the POI, including the front and rear of the device.

TA13.17 The tester shall describe and provide a costing for the most feasible attack to penetrate the ICC reader to make any additions, substitutions, or modifications to either the ICC reader's hardware or software, in order to determine or modify any sensitive data. The tester shall detail where costing information is based on testing or assumptions and provide justification for any use of assumptions rather than actual testing.

If an attack scenario can be developed that requires an attack potential of less than 20 (26 for SCRPs) per device for identification and initial exploitation or less than 10 (13 for SCRPs) for initial exploitation per device as defined in Appendix B, the vendor assertion cannot be verified. At least one attack scenario shall be presented, in a format consistent with the examples shown in Appendix B in this document. Calculations shall include evidence justifying particular rating levels as being appropriate.

The tester may perform any test needed to validate the attack scenario. The tester shall present evidence of the test methodologies followed and the validation results.

DTR A14 ICC Reader Construction and Slot Visibility

The opening for the insertion of the IC card is in full view of the cardholder during card insertion so that any untoward obstructions or suspicious objects at the opening are detectable.

The ICC reader is constructed so that wires running out of the slot of the IC reader to a recorder or transmitter (an external bug) can be observed by the cardholder.

Guidance:

The intent of requirement A14 is to make successful installation of PIN-disclosing bugs via the card slot infeasible. To meet this requirement the cardholder must have at least the ability to inspect the entry. And where the slot is neither positioned straight towards the cardholder nor upward facing —i.e., it is downward facing—a design has to meet the following criteria:

- *The ICCR slot entry area must be designed such that a cardholder has a full unlimited view of the housing surrounding the card slot opening. The card entry area should be extended to make it easier to observe the card slot area.*
- *The ICCR contacts must be strongly protected to prevent attachment of bug wires.*
- *There must not be any seams around the slot that can be used to hide wires.*
- *The ICCR slot internal sizes must not be sufficient to simultaneously insert two un-embossed cards and execute a transaction in order to minimize the likelihood of sufficient space for a bug.*
- *The maximum downward angle of the ICCR slot from horizontal should be no more than a maximum of 70 degrees.*
- *The installation guidance and security policy must stipulate the allowed installation height ensuring a sufficient view on the card slot entry area and the lab must validate that when the device is set at the minimum height the area around the slot is visible.*

TA14.1 The tester shall examine a test device to verify vendor assertions that the ICC reader's slot is in full view of the cardholder so that any untoward obstructions or suspicious objects at the opening are detectable. The construction of the device should be such that the entire slot opening is in full view of the cardholder prior to card insertion.

TA14.2 The tester shall provide a picture of the external area view of the customer ICC acceptor. The tester shall justify how this picture shows that the opening is in full view of the customer during a transaction, so that any objects within the slot would be clearly visible.

TA14.3 The tester shall note whether the slot is formed as one whole plastic part or is formed within the part line of two or more plastic parts.

TA14.4 If the POI is supplied with any add-on parts, such as privacy shields, stands, additional card readers, etc., the tester shall provide pictures of each of these parts showing how they attach to the POI, and confirm that each of these parts does not cover the ICC slot in any way.

TA14.5 The tester shall provide a picture of the internal view of the ICC acceptor and plastic casing. The tester shall note whether the acceptor is located so that it is difficult to pass or push a wire from the slot to some other external access point within the POI (such as a battery or SAM bay).

TA14.6 Where the slot is neither positioned straight towards the cardholder nor is upward facing, the tester shall provide illustration showing how the device meets the applicable guidance:

TA14.7 From the above information, the tester shall justify why it is not possible to pass a wire from the slot of the customer ICC acceptor without this wire being visible to the customer operating the POI.

B – Logical Security Derived Test Requirements

DTR B1 Self-Test

The device performs a self-test, which includes integrity and authenticity tests upon start-up and at least once per day to check whether the device is in a compromised state. In the event of a failure, the device and its functionality fail in a secure manner. The device must reinitialize memory at least every 24 hours.

Guidance

Firmware is considered to be any code within the device that provides security protections needed to comply with these requirements. The evaluating lab may require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.

The device must perform an internal self-test automatically at least once every day, in addition to power-up (excludes wake-up from hibernation mode). It is acceptable to perform firmware integrity checks before each PIN transaction as opposed to performing them at least once every 24 hours. Self-tests after several minutes of inactivity may also be used, rather than once every 24 hours, in addition to power-up self-tests.

For firmware that is rarely executed, such as a boot block, it is acceptable to perform an integrity and authenticity check immediately prior to each execution, rather than every 24 hours. However, note that all firmware must additionally be checked as part of the self-test performed at startup.

A memory re-initialization (security) cycle may last longer than 24 hours to allow the adjustment of the security cycle of the PIN entry device (maximum 24 hours duration) to the business cycle of an integrated POS system it may be connected to (maximum 24 hours duration). The firmware of the PIN entry device, during the cycles' adjustment processes, must not allow any security cycle to last longer than the combined maximum durations of the security cycle and the business cycle (48 hours). This must be included in the security policy for the device.

Firmware integrity tests may include techniques such as SHA-2 or equivalent. The hash must be either cryptographically protected using a key (for example, HMAC-SHA-2) or physically protected equivalent to a secret key. Authenticity testing must use cryptographic methods (MACs, digital signatures, or encryption). As such, an authenticity check will also confirm the integrity of the installed firmware, an additional integrity check is not necessary, but optionally may be additionally performed using a non-authenticated digest such as a CRC.

The self-tests shall include authenticated applications, as well as Open Protocols and Secure Reading and Exchange of Data code as applicable.

LRC, CRC, and other non-cryptographic methods and weak cryptographic methods (for example, SHA-1, MD5) are not allowed as the primary mechanism for either authentication or integrity checking.

The device controller is only in scope if it impacts one or more of the security requirements, for example, display prompt control.

Internal generation of a cryptographic signature is valid right after firmware update, assuming it complies with Requirement B2; it is also valid for devices that do not allow for software updates outside of the factory.

(continued)

Failing in a secure manner involves at least disabling any functionality of the device related to PIN handling and processing, including PIN entry and PIN encryption. More specifically, no sensitive data breach can occur as a consequence of device failure. For OP and SRED applications, failing in a secure manner involves disabling of all CHD processing functionality.

Chip-level code delivered with a component that cannot be configured, modified, or changed by any standard interface, and where an error cannot compromise the security of the device, does not need to be validated against Requirement B1. Examples may include smart card controllers, keypad controllers, or modem firmware.

TB1.1 The tester shall describe the boot chain of the POI. The tester shall include how initial machine code is loaded and executed by the processing elements, and how any subsequent firmware modules are sequenced, loaded, and executed, up to and including software modules used for PIN entry functions, account-data processing, and OP modules. The tester shall reference the relevant aspects of the asset flow.

TB1.2 The tester shall verify that the device performs self-tests upon start up and on a periodic basis at least once per day to check firmware and security mechanisms for signs of tampering, and whether the device is in a compromised state. The tester shall activate the self-test(s) and look for the result of the self-test(s) as shown by the device.

TB1.3 The tester shall verify that the device self-tests are able to detect failures and in doing so, fail in a secure manner. The vendor shall provide evidence of testing that confirms the relevant component's fails securely in the event of self-test failure.

TB1.4 For any self-test functions that are implemented by the built-in functions of the security processing elements, the tester shall detail what sources of information and testing have been used to validate that these processes are in place.

TB1.5 The tester shall review the source code of the POI to confirm what algorithms and keys are used to perform the self-test functions that are implemented by the firmware of the POI. The tester shall confirm that any register settings required to activate hardware-based self-test functions are correctly assigned.

TB1.6 The tester shall review the source code of the POI to confirm how it is ensured that the self-test process(es) are repeated every 24 hours, or prior to every PIN entry operation.

TB1.7 The tester shall note what methods are implemented to authenticate the cryptographic keys of the POI used for self-testing, to ensure that they have not been modified after loading.

TB1.8 The tester shall detail the processes performed by the POI if one or more of the self-test(s) fails. The tester shall confirm this through source-code review.

TB1.9 From the previous descriptions, the tester shall complete the following table indicating the process used to authenticate the firmware images during each stage of the booting process. The tester shall include all self-tests for all processing elements within the POI (as detailed in DTR A4). The tester shall adapt the table (for example, by adding columns or additional notes) as necessary, to present any additional information.

Boot stage	Algorithms and Key Sizes Used for Authentication	Area/Code/Registers Authenticated	Method and Frequency of Re-authentication	Action Performed if Failed

TB1.10 The tester shall confirm that the self-tests include validation of register settings relied upon for the security of the POI (for example, registers used to activate security features of the POI). The tester shall validate that these include checking of all features detailed and relied upon for compliance to the physical security requirements.

TB1.11 The tester shall perform testing to verify that the device reinitializes memory at least every 24 hours. The tester shall activate the mechanism and look for the result as shown by the device.

TB1.12 The tester shall review the source code of the POI to confirm how it is ensured that the re-initialization process is repeated every 24 hours or less.

TB1.13 The tester shall present sufficient evidence and/or references for the above, for compliance to this DTR.

DTR B2 Firmware Updates

The device must support firmware updates. The device must cryptographically authenticate the firmware and if the authenticity is not confirmed, the firmware update is rejected and deleted.

The update mechanism ensures security—i.e., integrity, mutual authentication, and protection against replay—by using an appropriate and declared security protocol when using a network connection.

Guidance

Firmware is considered to be any code within the device that provides security protections needed to comply with PCI requirements. Other code that exists within the device that does not provide security, and cannot impact security, is not considered firmware under PCI requirements. The evaluating lab may require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.

The authentication must not be performed by a component of lesser protection strength than the one for which the firmware/software is intended, OR the authentication must be performed by the target component of the firmware/software. For example, in an unattended device software/firmware for the EPP and the cryptographic module must be authenticated by these modules themselves only, while software for the application controller or the ICC reader may be authenticated by itself or by the EPP or by the cryptographic module.

The device must have the ability to accept firmware updates from a remote host—such as a terminal management system, using polling or similar techniques.

The firmware and application version numbers and the hardware version number must be shown on the display or printed during startup or upon request. This includes all modules addressed in testing, including SRED and Open Protocols.

The displayed firmware version number must represent all firmware in the device that is currently able to be executed.

- *If firmware blocks have independent version numbers, the version number display should include the version number of each firmware block.*
- *If a single version number is used, a documented process must be used to ensure the single version number is updated whenever changes are made to any of the firmware blocks in the device.*

This information shall be illustrated by photographic evidence provided in the evaluation report.

For SCRPs and SCRs intended for use with commercial-off-the-shelf devices—e.g., mobile phones and tablets—the SCRPs and SCRs must respond with their model name/number, hardware version, firmware version(s), and a unique device identifier to a query from the payment application on the COTS device.

(continued)

If done between physically and logically disparate components, it must use a secure channel as follows:

- *Each secure channel must provide mutual authentication to uniquely identify each component prior to exchanging sensitive data, as well as protect against MITM and replay attacks.*
- *Mutual authentication between the communicating components must be based on cryptography that aligns with Appendix E of this document, "Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms."*
- *Cryptographic keys used to establish secure channels between components and for data encryption must be unique, except by chance.*

TB2.1 The tester shall verify that the device supports firmware updates, including updates from a remote host, and that the vendor has a methodology for deploying updates remotely to the device as they become available (such as a Terminal Management System).

TB2.2 The tester shall determine by which component the authentication is performed.

TB2.3 The tester shall determine the level of protection for the external component involved in firmware/software updates and that the authentication of firmware updates is performed by a component of equal or greater strength.

TB2.4 The tester shall examine the vendor-supplied documentation to verify that the controls provide for unique accountability and utilize key sizes appropriate for the algorithm(s) in question. Examples of appropriate algorithms and minimum key sizes are stated in Appendix E, along with examples of acceptable hashing algorithms.

TB2.5 For each of the processing elements listed in DTR A4, the tester shall complete the following table. Where different parts of the code can be updated independently or if one part of the firmware cannot be updated, the tester shall ensure that this is detailed in the table as well. The tester shall reference the relevant aspects of the asset flow.

The tester shall adapt the table (for example, by adding columns or additional notes) as necessary, to present any additional information.

Firmware Element	Elements Used to Perform Authentication	Algorithms and Key Sizes Used for Firmware Authentication	Format of Authentication Block	Process Performed if Authentication Failed

If the method used for initial loading of the firmware differs from the method used for code update, provide additional details (including another of the tables above, if deemed necessary) of how initial code is loaded into the POI.

In the "Format of Authentication Block Column," include details on the format and padding of the authentication block.

TB2.6 The tester shall review the source code of the POI to confirm that the firmware-authentication methods are implemented correctly as noted above, and that the authentication is performed within the secure firmware of the POI. This evaluation activity should be focused at relevant security-critical sections of the source code, to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB2.7 If the POI allows for loading of multiple types of code (for example, firmware for security processor, firmware for magnetic-stripe reader encryption chip, application code, etc.), the tester shall detail how the various types of update images are differentiated from one another to prevent one type of image being incorrectly loaded into the wrong processing element/location. The tester shall ensure all authentication methods and image types are contained in the table above.

TB2.8 If (H)MAC method(s) are used for firmware authentication, the tester shall confirm through source-code review that the method used to compare the firmware-authentication block does not leak timing information (for example, the "C" memcmp() function is not used). This evaluation activity should be focused at relevant security-critical sections of the source code, to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB2.9 If a CBC MAC is used for firmware authentication, the tester shall detail what methods are used to mitigate vulnerabilities when authenticating variable-length data.

TB2.10 For each of the methods of authentication, the tester shall obtain a correctly authenticated firmware image and:

a) Confirm that it loads correctly into the POI. The tester shall detail the process involved in performing the loading.

b) Change a single bit in the authentication block for the entire image and confirm that this modified image is rejected when loaded into the POI.

c) Change a single bit in the firmware block of the image and confirm that this modified image is rejected when loaded into the POI.

TB2.11 The tester shall confirm how any public or private/secret keys are loaded into the POI during manufacturing. The tester shall specifically note whether any default values are installed (for example, default public certificates hard-coded into the firmware of the POI) and how it is ensured that these must be changed in deployed devices.

TB2.12 The tester shall verify and demonstrate that the device displays or otherwise makes available the firmware and application version numbers and the hardware version number during startup or upon request. This includes all firmware numbers for impacted requirements.

For SCRPs and SCRs intended for use with commercial-off-the-shelf devices, the tester shall confirm that the SCRPs and SCRs respond with their model name/number, hardware version, firmware version(s), and a unique device identifier to a query from the payment application on the COTS device. A secure channel as defined in the guidance must be used where disparate components are used.

TB2.13 Where a network connection is supported, the tester shall describe how a secure channel can be used for protection of software updates. Describe how the firmware-provided secure channel provides mutual authentication, integrity, and replay protection.

DTR B2.1 Application Authenticity

The firmware must support the authentication of applications loaded onto the terminal consistent with B2.

Guidance

Applications are considered to be any code that can be loaded onto the device that is not firmware. Other code that exists within the device that does not provide security, and cannot impact security, is not considered.

The authentication must not be performed by a component of lesser protection strength than the one for which the access is intended, OR the authentication must be performed by the target component.

If the device allows software application and/or configuration updates, the device cryptographically authenticates updates consistent with B2.

TB2.1.1 The tester shall determine by which component the authentication is performed.

TB2.1.2 The tester shall determine the rank of protection strength for the component involved in application authentication, including configuration updates and that the authentication is performed by an appropriate component.

TB2.1.3 The tester shall examine the vendor-supplied documentation to verify that the controls provide for unique accountability and utilize key sizes appropriate for the algorithm(s) in question. Examples of appropriate algorithms and minimum key sizes are stated in Appendix E, along with examples of acceptable hashing algorithms.

TB2.1.4 For each of the processing elements listed in DTR A4, the tester shall complete the following table. Where different parts of the code (for example, boot code, main firmware, etc.) can be updated independently, or if one part of the application cannot be updated, the tester shall ensure that this is detailed in the table as well.

The tester shall adapt the table (for example, by adding columns or additional notes) as necessary, to present any additional information.

Processing/ Application Element	Elements Used to Perform Authentication	Algorithms and Key Sizes Used for Application Authentication	Format of Authentication Block	Process Performed if Authentication Failed

If the method used for initial loading of the application differs from the method used for code update, provide additional details (including another of the tables above, if deemed necessary) of how initial code is loaded into the POI.

In the "Format of Authentication Block" column, include details on the format and padding of the authentication block.

TB2.1.5 The tester shall review the source code of the POI to confirm that the application authentication methods are implemented correctly as noted above, and that the authentication is performed within the secure firmware of the POI. This evaluation activity should be focused at relevant security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB2.1.6 If the POI allows for loading of multiple types of code (for example, application for security processor, application for magnetic-stripe-reader encryption chip, application code, etc.), the tester shall detail how the various types of update images are differentiated from one another to prevent one type of image being incorrectly loaded into the wrong processing element/location. The tester shall ensure all authentication methods and image types are contained in the table above.

TB2.1.7 If (H)MAC method(s) are used for application authentication, the tester shall confirm through source-code review that the method used to compare the application-authentication block does not leak timing information—for example, the “C” memcmp() function is not used. This evaluation activity should be focused at relevant security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB2.1.8 If a CBC MAC is used for application authentication, the tester shall detail what methods are used to mitigate vulnerabilities in this method when used to authenticate variable-length data.

TB2.1.9 For each of the methods of authentication the tester shall obtain a correctly authenticated application image and, if applicable, a software and/or configuration update, and:

a) Confirm that it loads correctly into the POI. The tester shall detail the process involved in performing the loading.

b) Change a single bit in the authentication block for the entire image and confirm that this modified image is rejected when loaded into the POI.

c) Change a single bit in the application block of the image and confirm that this modified image is rejected when loaded into the POI.

DTR B2.2 Signing

The vendor must provide a defined and documented process containing specific details on how any signing mechanisms must be implemented. This must include any "turnkey" systems required for compliance with the management of display prompts, or any mechanisms used for authenticating any application code. This must ensure:

- ***The signing process is performed under dual control.***
- ***All executable files are signed.***
- ***Software is only signed using a secure cryptographic device—e.g., smartcard—provided by the terminal vendor.***

Guidance

All executable files must be signed. By default, all other files should also be signed unless there is clearly documented justification why a signature is not required—i.e., the file cannot affect the security of the device.

Signing applies to any and all files that are executed or interpreted on the system (by either the firmware or the application). The firmware will have the ability to verify file signatures and will delete anything that fails verification.

TB2.2.1 The tester shall verify that the signing process can only be performed under dual control and that the software is only signed using a secure cryptographic device provided by the vendor.

TB2.2.2 The tester shall verify that any unsigned executable file cannot launch and is deleted by the device. The tester shall detail the device response.

TB2.2.3 The tester shall validate the documented justification for each unsigned file to ensure the file cannot affect the security of the device and that the justification is complete and correct.

TB2.2.4 The tester may perform any additional analysis necessary to validate the device's documentation. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid. The vendor shall provide any necessary test support to access and use the interfaces under test.

DTR B3 Differentiation of Entered PIN

The device never displays the entered PIN digits. Any array related to PIN entry displays only non-significant symbols—e.g., asterisks. If PIN entry is accompanied by audible tones, the tone for each entered PIN digit is indistinguishable from the tone for any other entered PIN digit.

Guidance

Any value or tone can be output as long as it cannot be used to determine PIN values. Using a different value or tone for different digit numbers or groups of numbers is not acceptable.

Any symbol can be output as long as it cannot be used to determine PIN values. Using a different symbol for different digit numbers or groups of numbers is not acceptable. Here is an example of symbol use that would NOT be allowed: 1=, 2=@, 3=%.*

A PIN-handling component of the device (for example, the PIN entry device) never outputs information to another component (for example, a display, speaker, or a device controller), allowing the differentiation of the PIN digits entered. Digit presses on touchscreen devices should never be displayed.

The evaluating lab shall require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.

TB3.1 The tester shall perform a test in which a PIN is entered to verify that the device does not output any digits of the PIN value. The tester shall note and report any characters, signals, or tones that are outputted. The device must display the same non-significant character for all PIN-entry uses. The tester shall provide an image capturing this.

TB3.2 If the device does not directly control the display, it must supply a suitable signal to indicate that a numeric key has been pressed and the value is stored inside the device. The tester shall determine the kind of signaling and to verify and describe how the signal information is not related to the digit entered.

TB3.3 The tester shall confirm through testing or source-code review which of the POI interfaces provide notifications of entered values during customer PIN entry. The tester shall include the display and logical interfaces for this investigation. This evaluation activity should be focused at relevant security-critical sections of the source code, to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB3.4 The tester shall review the API interface of the POI and detail any options provided with the PIN entry API that may be used to alter the notification(s) output during PIN entry.

TB3.5 If the POI implements a touchscreen for PIN entry, the tester shall verify that it does not provide visual indication of any of the digits entered—e.g., no “button highlight” is used.

TB3.6 The tester shall perform and describe a PIN entry process and monitor all of the POI interfaces during this process. The tester shall confirm and demonstrate that any notifications provided by the POI are consistent with expectations, as outlined above, and do not provide for determination of any of the entered PIN digits.

TB3.7 The tester shall describe the method used by the POI to provide audible feedback for customer PIN entry, detailing what method is used to control both the duration and frequency of the audible tone.

TB3.8 The tester shall connect an oscilloscope to the input of the sounding device of the POI and use microphones to monitor the signal for each of the ten numeric buttons. The tester shall perform signal analysis and signal processing as necessary to demonstrate that audible information determining key-press values is not leaked by the POI. The signals obtained should be sufficiently free from noise for the following analysis steps to be adequately performed. Perform multiple entries for each of the buttons. The tester shall attempt to associate the duration, tone, or any other attribute of the audible feedback of the POI to the actual button being pressed as enumerated below.

a) The tester shall confirm whether the duration for each button press is the same.

b) For devices where the sounding device frequency is externally controlled, the tester shall confirm whether the frequency for each button press is the same.

c) If either the frequency or duration differs between button presses, the tester shall state whether this is the case only when different buttons are pressed or remains true for when the same button is pressed multiple times. The tester shall show images and provide descriptions as necessary to support conclusions.

DTR B4 Clearing of Internal Buffers

Sensitive data shall not be retained any longer, or used more often, than strictly necessary. Online PINs are encrypted within the device immediately after PIN entry is complete and has been signified as such by the cardholder—e.g., via pressing the enter button.

The device must automatically clear its internal buffers of full track data (or chip equivalent) and sensitive authentication data is cleared when either:

- ***The transaction is completed, or***
- ***The device has timed out waiting for the response from the cardholder or merchant.***

Guidance

Vendor shall provide documentation of test results for inspections of internal buffers.

In terms of software or hardware architecture, encrypted immediately means when the cardholder signifies that PIN entry is complete, either by pressing an "enter" button or by entering the last digit of the PIN, the device does not perform any processes other than those required to encrypt the PIN.

Clear-text PINs must not exist for more than ninety seconds maximum from the completion of the cardholder's PIN entry. In all cases, erasure of the clear-text PIN must occur before the tamper-detection mechanisms can be disabled using attack methods described in A2.

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.

TB4.1 The tester shall verify that—and summarize how—the vendor has identified all data that is automatically cleared when the transaction is completed, and that all sensitive data is included. Passwords/authentication codes, clear-text PIN or account-data values, clear-text cryptographic keys, or key components outside of the crypto-processor are considered sensitive data.

TB4.2 The tester shall review the source code of the POI and confirm that—and summarize how—sensitive information is cleared from all storage locations after use, including local variables (before exiting the function) and registers. The tester shall detail the methods used to perform the erasure.

TB4.3 The tester shall detail the method used by the vendor to ensure that this buffer-clearing code/function cannot be removed by compiler optimizations or other means of code optimization, if employed by the vendor.

TB4.4 The tester shall detail the testing performed to verify that buffer-clearing code/function is robust. This requires assistance from the vendor and may involve, for example:

- Review of a small sample of compiled object code to validate that the code to clear the buffer remains in the compiled code;
- Extraction of memory from a special sample device after execution of the buffer-clearing code; or
- Confirmation that any compiler flags to ensure optimization are functioning as expected.

TB4.5 The tester shall confirm that and indicate how, via review, the vendor has the requirement for buffer clearing documented in their software-development practices documentation, including specific notes on how this should be done to prevent removal by compiler optimization, and that the correct implementation of this guide is reviewed as part of the firmware-verification process validated as part of DTR E2.

TB4.6 The tester shall perform any additional tests necessary to verify that all data is automatically cleared when either the transaction is completed or the device has timed out waiting for the response from the cardholder or merchant—for instance, by performing a partial simulated transaction to verify the behavior at time-out, or in general by entering the states that have been defined by the vendor under TB4.

DTR B5 Protection of Sensitive Services

Access to sensitive services requires authentication. Sensitive services provide access to the underlying sensitive functions. Sensitive functions are those functions that process sensitive data such as cryptographic keys, PINs, and passwords/authentication codes. Entering or exiting sensitive services shall not reveal or otherwise affect sensitive data.

Guidance

Authentication shall require dual-control techniques when entering sensitive information through a secure user interface, or cryptographic techniques when entering electronic data. The use of other techniques to access sensitive services results in the device being unable to use previously existing keying material. In all cases, the authentication values (passwords, authentication codes, or similar) on a given device must be different for each user.

A sensitive service (state) allows the execution of functions that are not available during normal use—for example, load a master key, alter device configuration, etc.

Key components entered manually constitute sensitive data during entry and the device shall not differentiate via sound or display the entry of different values.

TB5.1 The tester shall verify from vendor documentation—and summarize—that the vendor has identified all sensitive services, data and secure modes. Sensitive functions are those functions that process sensitive data such as cryptographic keys, PINs, and passwords/authentication codes. The tester shall verify from vendor documentation that sensitive services are authenticated and are entered, used, and exited securely and that mode transitions (for example, from operational to maintenance) do not reveal or otherwise affect sensitive information.

TB5.2 If access to sensitive services requires input by the keypad, the tester shall verify and describe that the protections for PIN data, such as the following, are also afforded to data entered while accessing sensitive services:

- Data inputs cannot be discerned from any displayed characters.
- Data inputs cannot be discerned by monitoring audible or electro-magnetic emissions.
- Sensitive data is cleared from internal buffers upon exiting the sensitive mode.

The testing shall include (but is not restricted to):

- Entering data while accessing sensitive services.
- Document review.

TB5.3 If mode transitions require input by a separate interface device, such as a key loader, the tester shall document the mechanism(s) and methodology used.

TB5.4 The tester shall detail in the table below all methods that can be used to load cryptographic keys into the POI. This shall include situations in which there is more than one loading method for any particular key, and in which different cryptographic keys may have different methods of loading. The tester shall include any APIs provided by the firmware that allow for the loading of cryptographic keys (reference the API list provided as part of DTR D2).

The items provided in the table below are for the purposes of example only:

Cryptographic key	Method of loading per TB5.5	Authentication
TMK	Components through keypad	Two seven-character passwords through keypad
	Encrypted under PKman	Provided by encryption
	Clear text through serial port	Operator authentication provided on key-loading device
Pkman	Embedded in firmware	Two eight-character passwords required to operate firmware loading

TB5.5 The tester shall verify that the loading of private and secret keys uses one or more of the following methods.

***Note:** EPPs and OEM PEDs intended for use in an unattended environment shall only support methods a, c, and d. SCRPs shall only support the loading of encrypted keying material.*

a) When entering clear-text secret keys through the keypad, they must be entered as two or more components and require the use of at least two passwords/authentication codes. The passwords/authentication codes must be entered through the keypad or else conveyed encrypted into the device. These passwords/authentication codes must either be unique per device (and per custodian), except by chance, or if vendor default, they are pre-expired and force a change upon initial use. Passwords/authentication codes that are unique per device can be made optionally changeable by the acquirer, but this is not required. Passwords/authentication codes are at least seven characters or an equivalent strength.

Entry of key components without the use of at least two separate passwords/authentication codes results in the zeroization of pre-existing secret keys, i.e., the invoking of the key-loading function/command causes the zeroization prior to the actual loading of the new key. For devices supporting multiple key hierarchies (for example, multi-acquirer devices), only the hierarchy (specific TMK and working keys) associated with the key being loaded must be zeroized.

b) For injecting clear-text secret or private keys from a key loader (which must be some type of secure cryptographic device), either the key loader or the device or both must require two or more passwords/authentication codes before injecting the clear-text key into the device.

***Note:** This may be the entire key—if components/shares, each component/share requires a separate password/authentication code.*

Passwords/authentication codes are at least seven characters or an equivalent strength. These passwords/authentication codes are entered directly through the keypad of the applicable device or are conveyed encrypted into the device. These passwords/authentication codes must either be unique per device (and per custodian), except by chance, or if vendor default, they are pre-expired and force a change upon initial use. Clear-text keys or their components/shares are never permitted over a network connection.

Injection of clear-text secret keys or their components/shares where the device does not itself require the use of at least two passwords/authentication codes for injection results in the zeroization of pre-existing secret keys. For devices supporting multiple key hierarchies (for example, multi-acquirer devices), only the hierarchy (specific TMK and working keys) associated with the key being loaded must be zeroized.

c) For encrypted values injected into the device, either from a key loader or from a network host, or via loading through the keypad, the ability of the device to successfully decrypt the value and use it is sufficient. In this case, the loading of the key-encipherment key would have been done under dual control, for example in a) and b) above.

d) Remote key-loading techniques using public key methods requires compliance with PCI defined criteria for key sizes and mutual authentication between host and device. For devices generating their own key values, the generation process must meet the criteria defined in Appendix D, “Configuration and Use of the STS Tool,” and validation that appropriate key sizes are used. The protocol must meet the criteria stipulated in Appendix A of the PCI PIN Security Requirements. See below for further information.

The tester shall include a reference to the method used in each of the rows in the table above.

TB5.6 The tester shall justify why each of the methods that can be used to load cryptographic keys enforces both dual control and split knowledge.

TB5.7 The tester shall verify that the loading of all private or secret keys can be performed without using plaintext key injection (as required by PCI PIN). If an asymmetric key-loading technique which does not support mutual cryptographic authentication is used for loading the initial keys, a secure room as defined in PIN Security Requirements 32-9 must be used. If devices generate key pairs and have the public key signed, this process must occur in this same secure environment.

Note: *This does not apply to key components entered into the keypad of a secure cryptographic device.*

TB5.8 The tester shall detail any further sensitive services provided by the POI that have not been addressed as defined above. This must include the disabling or alteration of any systems or functions relied upon by the POI to meet the PTS requirements. Examples include but are not limited to changing SRED encryption modes or altering system time that may be used to verify certificate validity (or used for other system security services).

TB5.9 The tester shall confirm that and describe how any entry of sensitive information through the keypad of the POI is protected to the same extent as customer PINs, conforming to all relevant requirements (for example, A1, A2, A4, A5 and A6).

TB5.10 For any asymmetric key-loading methods, the tester shall provide a message-flow diagram showing the contents of each data packet exchanged during the asymmetric key-loading process, including any authentication parameters, freshness indicators, and padding. From this diagram, for each asymmetric key-loading method, the tester shall provide the following justifications:

a) Why any packets that do not have an authentication parameter attached cannot be used as part of a man-in-the-middle attack.

b) Why any freshness indicators are sufficient to prevent replay of any part of the message that would result in the re-loading of a previously installed cryptographic key.

c) Why it is not possible for any party, other than the “owner” of the existing public key, to change any public key in the device once it has been initially loaded.

d) Why the protocol provides mutual authentication of the POI and host before any cryptographic key is used for a security-sensitive process (such as cryptographic key loading).

TB5.11 Where public keys are loaded into the POI in a non-secure environment (for example, during firmware loading), the tester shall justify how dual control is maintained and alteration or manipulation of the public key value(s) is not possible.

Note: *This applies to the loading of initial high-level—e.g., root—keys. Where those keys are embedded in the firmware, they will leverage the mechanisms used for the firmware as required under E5. Other public keys that are loaded after the device has received its FW can be authenticated using cryptographic mechanisms (like a certificate or a MAC), and the loading can happen in a non-secure area without compromising security.*

TB5.12 The tester shall detail any other sensitive services offered by the POI and detail the authentication provided for these services. The tester shall justify how this ensures dual control is provided for all sensitive services.

TB5.13 The tester shall validate that—and describe how—all passwords/authentication codes implemented to provide dual control are at least seven characters or an equivalent strength.

TB5.14 The tester shall attempt to load cryptographic keys or components into the POI without changing the default values of the passwords/authentication codes. The tester shall detail the results. The requirements of this DTR are not met if this can be done.

TB5.15 The tester shall attempt to set the passwords/authentication codes in the POI so that two or more of the passwords/authentication codes in the same device have the same value. The tester shall detail the results. The requirements of this DTR are not met if this can be done.

TB5.16 The tester shall attempt to set the passwords/authentication codes of the POI to a value that is less than seven characters or an equivalent strength. The tester shall detail the results. The requirements of this DTR are not met if this can be done.

DTR B6 Sensitive Services Limits

To minimize the risks from unauthorized use of sensitive services, limits on the number of actions that can be performed and a time limit shall be imposed, after which the device is forced to return to its normal mode.

Guidance

The evaluating lab may require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.

TB6.1 The tester shall verify the following:

- The vendor has provided a rationale for the value chosen as a limit on the number of actions and the time limits imposed.
- The vendor has provided a rationale as to how the limits minimize the risks from unauthorized use of sensitive services.

TB6.2 The tester shall verify the limits placed on the number of actions by causing the device to access sensitive services and attempting to exceed the limit. Once the limit is exceeded the tester shall verify that the device has returned to its normal mode.

TB6.3 Referencing the sensitive services outlined in the previous requirement, the tester shall detail the functional limits provided for each of these services.

TB6.4 For any password-entry modes, the tester shall detail how the functional limits ensure that any arbitrary password/authentication code guess has less than a 1/10000000 chance of success, and any multiple attempts within a one-minute period have a less than 1/10000 chance of success of correctly providing the password/authentication code value. The tester shall detail the testing performed to validate these functional limits.

TB6.5 For any password-entry modes, the tester shall confirm through source code examination that the method of validating the password/authentication code value is not vulnerable to a timing attack (for example, a standard “strcmp” or “memcmp” function is not used), or that the functional limits are set to values that prevent utilization of any leaked timing information. This evaluation activity should be focused at relevant, security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB6.6 For all sensitive services, excluding loading of encrypted key values but including loading of clear-text key value from an external key-loading device, the tester shall confirm that a timer is implemented that will return the POI to a non-sensitive mode after a maximum of 15 minutes, regardless of input or continued activity.

For sensitive services requiring the input of two or more passwords/authentication codes, the maximum timer must be started upon entry of the first password/authentication code digit and include the time taken to both enter the password/authentication code and perform the service that password/authentication code authenticates (for example, entry of a single key component, or access to a sensitive menu function). If the POI separates the entry of each component such that a single password/authentication code is entered prior to each component value, it is not necessary for the POI to implement the timer between each password/component entry operation.

Validation of the maximum timer value may involve testing of only one of the sensitive states if source-code review can confirm that the same code is used for all sensitive states. The tester shall detail the specific method of testing used, and how these results ensure that it is not possible to maintain a sensitive state for more than 15 minutes of elapsed time.

TB6.7 For all sensitive services requiring the input of passwords/authentication code and key components into the POI keypad, the tester shall confirm that an inactivity time-out is implemented such that if a button is not pressed every 60 seconds, the device will exit the sensitive state.

Validation of the time-out value may involve testing of only one of the sensitive states if source-code review can confirm that the same code is used for all sensitive states. The tester shall detail the specific method of testing used and how these results ensure that a lack of any input for a period in excess of 60 seconds will result in the exit of the sensitive state.

DTR B7 Random Numbers

If random numbers are generated by the device in connection with security over sensitive data, the random number generator has been assessed to ensure it is generating numbers sufficiently unpredictable.

Guidance

Unpredictability of random numbers is as important as distribution. The implementation shall ensure that seeding or initializing the random number generator at startup cannot be abused to intentionally reproduce a given random value or sequence.

The device shall be capable of meeting the statistical tests of NIST SP PUB 800-22. *See Appendix D.*

Usages include use for the EMV unpredictable number.

The scope of this requirement includes random numbers that are generated in connection with meeting requirements applicable for Open Protocols and Secure Reading and Exchange of Data.

Random numbers generated for use in the creation of cryptographic keys require a strong source of entropy in order to prevent predictability. Other uses, such as the generation of random nonces for the prevention of replay attacks when using asymmetric techniques to distribute symmetric keys may only require the guarantee of uniqueness.

Pseudorandom number generator (PRNG) designs (also known as deterministic random bit generator, or DRBG) from NIST SP800-90A *or* ANSI X9.82 *shall be used. Specifically, HASH_DRBG, HMAC_DRBG, or CTR_DRBG. DEA and 2-key TDEA, as well as DUAL_EC_DRBG, are not acceptable for use in a DRBG.*

Source code will be required to validate this requirement.

The evaluating lab may require assistance from the vendor to make a systematic review of relevant security functions.

B7 is mandatory for the SCRP approval class in order to provide a source of entropy for external payment applications. It is a requirement that any random numbers used on the COTS device for security purposes must be seeded from a value provided from the RNG on an SCRP.

TB7.1 The tester shall detail the method used by the POI to generate random numbers, including any seed values used, hardware systems, and software-based DRBGs.

The tester shall outline the process used by the vendor to ensure that any secret values relied upon for random number generation (such as seed values, or keys used in DRBGs) are sufficiently random, and unique per POI.

The tester shall justify why the method used by the POI to generate random numbers is robust.

TB7.2 The tester shall confirm that the process outlined in TB7.1 includes a method to provide entropy (the seed length shall be at least 256 bits) from a hardware-based source, as well as a software-based “whitening” process such as a DRBG to remove any bias in the hardware-based system. The tester shall provide a justification that this method ensures that sufficient entropy is provided for all uses of the random number generator. For example, the tester might cite that when the RNG is used for generating EMV Unpredictable Numbers it applies the EMV-approved unpredictable number generation algorithm.

The tester shall ensure that initializing and/or seeding of the RNG cannot be abused to intentionally reproduce a given random value or sequence.

TB7.3 The tester shall list all security services implemented within the POI that require or rely upon the use of random data. This may include generation of padding data (for example, for use in certificates, key blocks, EMV Unpredictable Numbers, or offline encrypted PIN blocks), generation of cryptographic keys, randomization of the keypad scanning algorithm, etc.

TB7.4 The tester shall review the source code of each of these services and confirm that they correctly utilize the random number generator reviewed in this requirement. This evaluation activity should be focused at relevant security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB7.5 The tester shall obtain at least 128MB of random data from the POI under test. This data may be supplied directly by the vendor. The tester shall detail the method used to generate this data and justify why this sufficiently replicates the way in which the RNG will be used by the POI. The tester shall pass the 128MB of data through the NIST STS test program, and detail the results, indicating pass and fail results and how these demonstrate compliance to this DTR. In some situations, the data set will produce minor fail cases on individual tests, notwithstanding sufficient random generation. In this case, the tester shall repeat the tests using a second data set to demonstrate the STS tests genuinely pass. See Appendix D.

DTR B8 Exhaustive PIN Determination

The device has characteristics that prevent or significantly deter the use of the device for exhaustive PIN determination.

Guidance

The following are examples of techniques that may be used to prevent an exhaustive PIN determination attack, such as one using electromechanical solenoids to depress the keys so as to try all possible PINs until the ciphertext produced equals the ciphertext recorded when the device was in operational use:

- *Use of a unique-key-per-transaction technique. (Prevents the attack.)*

  Note: *Master/session is NOT a unique-key-per-transaction technique.*

- *Preventing the entry of the PIN through other than the keypad and limiting the rate at which the device will encrypt PINs to the average (for example, over 120 transactions) of one per 30 seconds. The intent of the requirement statement is that for **any** 120 consecutive transactions, the average time between encryptions for a specific PIN entry is approximately 30 seconds. (Deters the attack.)*
- *The device is exclusively used for offline PIN and the ICC reader is integrated into the PIN entry device.*
- *The exclusive use of ISO PIN-block formats 3 and/or 4 whereby each PIN is enciphered using a unique except by chance random pad of characters with permissible values ranging from 0000 to 1111 depending on the format may be used to prevent exhaustive PIN determination.*

The average time delay between encryptions should be calculated for exhaustive attack to determine a single PIN value and is NOT averaged over attacks on multiple PINs.

Offline devices that do not have the PIN entry device and the ICC reader integrated into the same secure module, and which are using ISO format 0 and are not using a unique key per transaction for the conveyance of the PIN from the point of entry to the ICC reader, must comply with this requirement.

The evaluating lab may require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.

TB8.1 The tester shall note whether the POI only supports key-management methods that ensure a unique key is used for each PIN encryption. If this is the case, the tester shall justify why the methods used to ensure a unique key are sufficient, and no further testing is necessary for this requirement.

TB8.2 The tester shall note whether the POI only supports for online PIN ISO format 4 or ISO format 3 PIN blocks. If so, the tester shall review the source code of the POI to confirm that the padding used on these PIN blocks is generated by the random number generator validated under Requirement B7. If this is the case, no further testing is necessary for this requirement.

TB8.3 If the previous areas of testing were negative, or have not been conducted, the tester shall detail the method used by the POI to ensure that it is not possible to encrypt more than 120 PINs in less than one hour of elapsed time.

TB8.4 The tester shall review the source code of the POI to confirm that—and summarize how—the implementation matches the vendor attestations, and that the method is implemented for all PIN entry methods and key-management types (excluding those noted above as being exempt from this requirement if confirmed to be valid). This evaluation activity should be focused at relevant security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB8.5 Using a functional sample device, the tester shall attempt to perform at least 120 PIN entry operations. The tester shall note the time taken and the response of the POI when the final transaction is performed, and whether another transaction is possible before an elapsed time of one hour from the entry of the first PIN.

TB8.6 Using a functional sample device, the tester shall attempt to perform 120 PIN entry operations in less than one hour. If the device blocks attempts to perform PIN entry, the tester shall remove the power from the device. Upon re-powering the device, the tester shall attempt to perform another PIN entry operation and note whether this is possible. The tester shall describe the device's behavior under this test. If the tester can perform another (121st) PIN entry, the device fails.

TB8.7 If the method used to prevent the entry of more than 120 PINs within an hour utilizes the real-time clock of the POI, the tester shall detail what methods are available to change the value of this clock and justify why these methods do not allow for the bypassing of the PIN entry rate limiting.

DTR B9 Key Management

The key-management techniques implemented in the device conform to **ISO 11568** ***and/or*** **ANSI X9.24.** ***Key-management techniques must support key blocks as defined in DTR B9.***

Guidance

POI devices used for IC card acceptance must have chipsets that support ECC and are enabled for use.

ECC functionality should support standard prime curves (at minimum the NIST recommended curves P-256 and P-521) and should include scalar multiplication of a point on the curve and prime field arithmetic (including exponentiation with full size exponent).

Symmetric key components shall be combined via either XOR'ing of full-length key components or via implementation of a recognized secret-sharing scheme, for example, Shamir. Private key components shall be combined using a recognized secret-sharing scheme. Devices must implement unique secret and private keys for any function directly or indirectly related to PIN or account-data protection. The basic rule is that any private or secret key resident in the device that is directly or indirectly used for PIN protection whose compromise would lead to the compromise of the same key in another device must be unique per device. For example, this means not only the PIN-encryption key(s), but keys that are used to protect other keys, firmware-update and authentication keys, and display prompt control keys. As stated in the requirement, this does not apply to public keys resident in the device.

When a check value is generated for a key or key component, it shall be generated by a cryptographic process such that all portions of the key or key component are involved in generating the check value.

Clear keys or clear-key parts must not be loaded using the service module.

A device may include more than one compliant key exchange and storage scheme.

Devices must support the ANSI X9.143 key-derivation methodology for TDES keys, and for AES keys must support the X9.143 methodology and/or the ISO 20038 methodology. X9.143 are recognized as interoperable methods for both TDEA and AES. ISO 20038 is recognized as an interoperable method for AES. The X9.143 Key Variant (Calculation) methods are no longer allowed.

In either case, equivalent methods can be used where subject to an independent expert review and said review is publicly available as described below. Other methods may additionally be supported where required by legal or regulatory requirements in specific markets but can no longer be supported in lieu of the aforementioned.

Any equivalent method must include the cryptographic binding of the key-usage information to the key value using accepted methods. Any binding or unbinding of key-usage information from the key must take place within the secure cryptographic boundary of the device. For all methods used, the encrypted key and its attributes in the key block shall have integrity protection such that they cannot be modified without detection. Modification includes, but is not limited to:

- *Changing or replacing any bit(s) in the attributes or encrypted key*
- *Interchanging any bits of the protected key block with bits from another part of the block*

Documentation must be provided demonstrating how the methodology meets these criteria.

(continued)

Equivalent methods must be subject to an independent expert review, and said review must be publicly available:

- *The review by the independent expert must include proof that in the equivalent method the encrypted key and its attributes in the key block have integrity protection such that it is computationally infeasible for the key to be used if the key or its attributes have been modified. Modification includes, but is not limited to:*
  - *Changing or replacing any bit(s) in the attributes or encrypted key*
  - *Interchanging any bits of the protected key block with bits from another part of the block*
- *The independent expert must be qualified via a combination of education, training, and experience in cryptology to provide objective technical evaluations that are independent of any ties to vendors and special interests. "Independent expert" is further defined below.*
- *The PTS laboratory will validate that any device vendors implementing this methodology have done so following all guidelines of said evaluation and peer review, including any recommendations for associated key management.*

An Independent Expert possesses the following qualifications:

- *Holds one or more professional credentials applicable to the field—e.g., doctoral-level qualifications in a relevant discipline or government certification in cryptography by an authoritative body (e.g., NSA, CES, or GCHQ); and*
- *Has ten or more years of experience in the relevant subject; and*
- *Has published at least two articles in peer-reviewed publications on the relevant subject, or*
- *Is recognized by his/her peers in the field—e.g., awarded the Fellow or Distinguished Fellow or similar professional recognition by an appropriate body, e.g., ACM, BCS, IEEE, IET, IACR; and*
- *Subscribes to an ethical code of conduct and would be subject to an ethics compliance process if warranted.*

Independence requires that the entity is not subject to control, restriction, modification, or limitation from a given outside source. Specifically, independence requires that a person, firm, or corporation who holds itself out for employment as a cryptologist or similar expert to more than one client company is not a regular employee of that company, does not work exclusively for one company, and where paid, is paid in each case assigned for time consumed and expenses incurred.

***Note:** Multiple individuals who collectively possess the necessary expertise who meet the independence criteria can be used. When using a group approach, each individual must have at least 10 years of experience in the relevant subject and must subscribe to an ethical code of conduct.*

This does not imply that the device must enforce X9.143 or an equivalent scheme, but it must be capable of implementing such a scheme as a configuration option.

These methods must be used for key loading whenever a symmetric key—e.g., AES or TDES—is loaded encrypted by another symmetric key. This applies whether symmetric keys are loaded manually—i.e., through the keypad—using a key-injection device, or from a remote host. It does not apply when clear-text symmetric keys or their components are loaded using standard dual control techniques.

(continued)

This does not imply that the device must support X9.143 for TDES and AES or ISO 20038 for AES or an equivalent methodology between the device and an external ICC reader, but it optionally may do so. The device may also optionally support X9.143 for TDES and AES or ISO 20038 for AES or an equivalent methodology for the storage of keys encrypted under a symmetric key.

The evaluating lab may require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.

The use of variants is not allowed for AES keys.

AES keys can only be:

- *Loaded using asymmetric keys of equivalent or greater strength—note a specific exception is allowed for the use of 2048 RSA keys to encrypt 128 bits AES keys for remote key distribution as described in ANSI X9.143 using asymmetric keys as delineated in PIN Security Requirement 10; or*
- *Encrypted or derived by another AES key of equal or greater strength; or*
- *Manually loaded using dual-control techniques; or*
- *Internally generated using a random number generator compliant with B7.*

Note that other public key techniques such as Diffie Hellman or Elliptic Curve must be used to convey AES keys greater in strength than 128 bits.

Remote key distribution using asymmetric techniques shall employ mechanisms to protect against man-in-the-middle attacks and the hijacking of PIN-acceptance devices. Acceptable techniques include:

1. *For devices under a PKI hierarchy that facilitates more than one acquirer—e.g., a hierarchy under a PIN-acceptance device vendor's root—the PIN-acceptance device can be forced to bind to a specific transaction-processing host's certificate, and not accept commands digitally signed by any other hosts. This is frequently done at initialization of a new PIN-acceptance device, and subject to unbinding techniques as noted below.*
2. *The acquirer KDH public key can be loaded only once and requires a factory return (preceded by a zeroization of acquirer keys function) to put the device back to ready state.*
3. *An acquirer-specific PKI hierarchy can be implemented. For this scenario, because of the rigor of criteria for operating a Certification Authority, it is best to have the PIN-acceptance device vendor operate the hierarchy or use a company that provides professional Certification Authority services.*
4. *Certificate Revocation Lists can be distributed to the device to identify compromised key-distribution hosts. This requires that device vendors maintain and distribute the CRLs for KDH keys that are part of their remote key distribution PKI. It further requires that the CRLs have a lifetime not to exceed one week to minimize the exposure window. Additionally, it requires that the device cease processing if it does not possess a valid unexpired CRL.*

Devices that are "bound" as stated above shall support a technique for decommissioning for unbinding from a specific host. Decommissions, such as sending a new host's certificate to replace the existing host's certificate without authentication, are not acceptable. Any remote decommissioning must require cryptographic techniques and be specific per PIN-acceptance device. Acceptable techniques include:

1. *The existing bound host can digitally sign an "unbind" command to the PIN-acceptance device, that when validated returns the PIN-acceptance device to its original unbound state.*

(continued)

2. *In situations where the bound host's private key is not available to sign the command, or similar scenarios, a forced decommission may occur. However, any such decommission done remotely requires a cryptographic (digital signature, MAC, etc.) technique and must be unique per PIN-acceptance device.*
3. *Decommissions may also be done manually directly at the device, using system administration menus that authenticate users via PINs, passphrases, etc.*

Other acceptable techniques include those stated in ANSI TR-34.

In all cases of decommissioning, the existing acquirer-related keys must be zeroized as a result of the decommission.

In the event of a permanent device decommissioning, the device may be tampered which must result in the zeroizing of all private and secret keys.

SCRPs shall provide for the use of cryptographic keys stored within the SCRP to provide security to the provisioning process. This process loads cryptographic keys and other data to an external system (such as a payment application on a mobile phone) upon initial configuration as described in DTR B8: Secure Provisioning of the PCI Software-based PIN Entry on COTS DTRs.

POI devices supporting remote key loading using asymmetric techniques cannot use the same key pair for both authentication and encryption purposes. Furthermore, these key pairs are only allowed for use in connection with key loading.

As stated in the PCI PIN Security Requirements*, RSA keys encrypting keys greater in strength than double-length TDEA keys shall use a modulus of at least 2048 bits.*

Fixed key (either AES or TDEA) shall not be supported for account-data or PIN encipherment.

The software library/chipset used to generate asymmetric keys shall be identified in the Asset Flow Diagram to help safeguard against weak key-generation algorithms.

TB9.1 The tester shall determine from vendor documentation the key-management technique used for firmware and application updates. Symmetric key techniques must include the use of Unique Key(s) per device.

TB9.2 The minimum key sizes and parameters for the algorithm(s) in question that must be used for key transport, exchange, or establishment are stated in Appendix E.

If a public-key technique for the remote distribution of symmetric secret keys related to PIN or account-data encryption is used, it must:

a) Use public and private-key lengths that are deemed acceptable for the algorithm in question (for example, 2048-bits minimum for RSA).

b) Use key-generation techniques that meet the current ANSI and ISO standards for the algorithm in question.

c) Provide for mutual device authentication for both the host and the device, including assurance to the host that the device actually has computed (or actually can compute) the session key and that no entity other than the device specifically identified can possibly compute the session key.

TB9.3 The tester shall determine from vendor documentation all storage and usage locations for each key for example, ROM, external RAM, EPROM, processor chip, etc., and list the details in a key summary table. The tester shall reference the relevant aspects of the asset flow.

TB9.4 The tester shall determine from vendor documentation how (for example, active or passive erasure) each key is destroyed for all device states (power-on, power-off, sleep mode) and list the details in a key summary table.

TB9.5 The tester shall note all acquirer-based key-management systems supported by the POI, defining them as one of *ANSI X9.24* DUKPT or Master/Session key management.

TB9.6 For systems that support Master/Session key management, the tester shall define what (if any) standard this key management complies with (for example, where the POI implements country specific key management). The tester shall note the name, version, and authoring party of this standard.

TB9.7 The tester shall review the API guide and operational manual of the POI and confirm that this does not detail any other key-management schemes or methods that the POI may use. If the POI supports extensible key management for use with non-PIN transactions, the tester shall justify what prevents this from being used with PCI-based PINs, using extracts from the vendor documentation to support this claim. The tester shall provide references to all extracts used.

TB9.8 If the POI implements an ICC reader, the tester shall detail which EMV-based keys are supported by the firmware. If EMV functions are provided by the application, the tester shall detail how offline PIN functions are provided, including how the PIN key of the customer ICC card is authenticated. The tester shall justify how the methods used are sufficient to prevent the application from accessing clear-text PINs.

TB9.9 The tester shall detail any other types of key management or cryptographic keys used by the POI. This should include any keys used for firmware or application authentication, self-testing, boot strapping, remote key injection, local key injection, dual control, etc.

TB9.10 The tester shall provide a key table for the POI, accurately including all of the keys and key-management methods outlined above. The key table shall include all PIN- or SRED-related keys used in the device, including but not limited to acquirer keys, software authentication keys, and storage of keys within the device. Cryptographic keys used for Open Protocol security must be listed in DTR D5. Keys that are not considered to be PIN or SRED security-related may either be listed in the table with a note or mentioned in a statement detailing how they cannot affect PIN or SRED security. The tester shall confirm that the key table is accurate and complete.

Key Name	Purpose/ Usage	Algorithm	Size (Bits)	Generated by:	Form Factor in which Key is Loaded to Device	Number of Available Key Slots (Registers)	Unique per Device/ Acquirer/ Vendor-specific/ Other (describe)	Location/ Storage Area	How Key is Identified by the Device so it is used only as Intended
Terminal Master Key (TMK)	Encryption of working keys (PEK, MAC) for down-line transmission to the device	TDES	128	Acquirer	2 or 3 clear-text components	One	Device		Designated Key Register
MAC Key	Message authentication	TDES, DES	128 or 64	Acquirer	Enciphered under the TMK	Two	Device		X9.143 Key Derivation Binding Method
PIN-encryption Key (PEK)	PIN encipherment for online PIN	TDES, DES	128 or 64	Acquirer	Enciphered under the TMK	Two	Device		X9.143 Key Derivation Binding Method

Key Name	Purpose/ Usage	Algorithm	Size (Bits)	Generated by:	Form Factor in which Key is Loaded to Device	Number of Available Key Slots (Registers)	Unique per Device/ Acquirer/ Vendor-specific/ Other (describe)	Location/ Storage Area	How Key is Identified by the Device so it is used only as Intended
IPEK	Initial DUKPT Key	TDES	128	Acquirer	Clear text from key-injection device	One	Device		Designated Key Register
DUKPT PEKs (Future Keys Register)	PIN encipherment for online PIN	TDES	128	Acquirer	Derived originally from IPEK	Up to 21 Future Keys	Device		Designated Key Register
KPT	PIN encipherment between EPP and IC card reader	TDES	128	Acquirer	2 or 3 clear-text components	One	Device		Designated Key Register
Payment Scheme (Certification Authority) Public Keys—e.g., EMV	Authentication of issuer key from IC card	RSA	Varies	Payment Schemes	EMV Public Key Certificate	Six per payment schemes – three payment schemes	Payment scheme-specific		Designated Key Register
Manufacturer Firmware Authentication Root or Sub-CA Public Key	Authentication of firmware updates as part of a certificate chain to the manufacturer root key	RSA	2048	Manufacturer	Certificate signed with manufacturer's private key	One	Vendor-specific		Designated Key Register
Manufacturer Authentication Root or Sub-CA Public Key	Authentication of acquirer-signed applications as part of a certificate chain to the manufacturer root key	RSA	2048	Manufacturer	Certificate signed with manufacturer's private key	One	Vendor-specific		Cryptographic Authentication under Vendor PKI
Acquirer's Application Public Authentication Key	Authentication of acquirer-signed applications as part of certificate chain to manufacturer root key	RSA	2048	Acquirer	Public Key Certificate signed by manufacturer root or sub-CA private key	One	Acquirer		Cryptographic Authentication under Vendor PKI
Manufacturer Authentication Root or Sub-CA Private Key	Signing firmware updates or the acquirer application signing public key	RSA	2048	Manufacturer	Managed at manufacturer's secure facility under dual control	One	Vendor-specific		N/A
Acquirer's Application Private Authentication Key	Signing application updates	RSA	2048	Acquirer	Managed at acquirer's secure facility under dual control	One	Acquirer		N/A

***Note:** The "Size" column must note both the maximum and minimum sizes that can be used for that key.*

TB9.11 Using the key table as a reference, the tester shall note which keys are actively erased by the POI during a tamper event, and which keys are not erased but instead rely upon the erasure of a KEK to prevent their subsequent misuse.

TB9.12 Using the key table as a reference, the tester shall confirm that all secret and private keys, including those used for account-data encryption, are unique per device, and what method is used to ensure this is the case.

TB9.13 Using the key table and API guide as a reference, the tester shall note which keys can be loaded by applications in clear text.

TB9.14 Using the table of sensitive-information storage from Requirement A4 and the key table above, the tester shall confirm:

a) No key is encrypted under a key of lesser strength—this includes KEKs, transport key (with the exception for 2048 RSA keys as noted in the guidance above), and storage keys. Where AES working keys are supported, such as for use with AES-based PIN blocks, the key hierarchy for those working keys (including the POI storage key) must not implement TDEA keys—i.e., must implement AES keys of equal or greater strength than the keys they protect; and

b) Clear-text cryptographic keys are not stored encrypted under bulk data-encryption keys (such as keys used to encrypt external memory).

TB9.15 The tester shall detail any ways in which the POI generates keys from other keys and justify why these are valid key-generation functions as required by ISO11568 and ANSI X9.24.

TB9.16 The tester shall note whether the POI generates any keys using an internal random number generator. The tester shall confirm through source-code review that these keys are generated using the same process validated under Requirement B7. This evaluation activity should be focused at relevant security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB9.17 The tester shall detail the method used by the POI to confirm that no one key can take the same value as any other key within the POI. Through source-code review confirm the following:

a) The method used does not provide a potential timing attack on the POI—for example, by using a standard C “memcmp()” function to compare all keys.

b) If key check values (KCVs) are used for this purpose, the KCVs stored are limited to values as defined in TB9.18 or they are never output from the POI.

c) The method used does not rely on the check digits—e.g., mod 10 calculation—of a (T) DES key as part of the key comparison.

This evaluation activity should be focused at relevant security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB9.18 Referencing the POI API, user guides, and other documentation, the tester shall confirm that it is not possible to output a KCV value except as noted below.

Note: *Check values may be computed by two methods. TDEA may use either method. AES must only use the CMAC method. In the first method, check values are computed by encrypting an all-binary zeros block using the key or component as the encryption key, using the leftmost n-bits of the result, where n is at most 24 bits (6 hexadecimal digits/3 bytes). In the second method the KCV is calculated by MACing an all-binary zeros block using the CMAC algorithm as specified in ISO 9797-1 (see also NIST SP 800-38B). The check value will be the leftmost n-bits of the result, where n is at most 40 bits (10 hexadecimal digits). The block cipher used in the CMAC function is the same as the block cipher of the key itself. For example, a TDEA key or a component of a TDEA key will be MACed using the TDEA block cipher, while a 128-bit AES key or component will be MACed using the AES-128 block cipher.*

TB9.19 The tester shall note what methods are implemented to authenticate the cryptographic keys of the POI to ensure that they have not been modified after loading.

TB9.20 The tester shall detail the key-block method(s) supported by the POI device, providing details on exact contents of any header, payload, padding, etc. as well as noting which parts are encrypted and which are included in any authentication calculation. If *ANSI X9.143* key-derivation method or *ISO 20038* is not used, the tester shall cite the publicly available independent expert review to justify why the method used provides the same level of security. Specifically, the tester shall note how the key-block method supports each of the following properties:

- Key confidentiality
- Key integrity
- Key purpose
- Algorithm used for the key
- Key length

TB9.21 The tester shall confirm that any key-block protection key can only be used for that purpose and cannot be used as a “generic” master or working key, as part of a non-key-block key-management scheme.

TB9.22 For any methods that rely on the use of full-length TDES key components for enforcing split knowledge, the tester shall attempt to load all but one of the components as an all-zero value. If this does not succeed, the tester shall attempt to load a zero-value component where the parity bits have been modified so that the actual value of the component entered is not composed entirely of zeros. For key shares, the tester shall use the same value for all but one share to perform the aforementioned.

The requirements of this DTR are not met if it is possible to load a key where the value of that key can be known through knowledge of only one component. The findings of these tests shall be explained.

TB9.23 The tester shall confirm that if the device supports remote key loading using asymmetric techniques that it utilizes an acceptable method to protect against man-in-the-middle attacks and the hijacking of payment-acceptance devices.

TB9.24 The tester shall confirm that for a device supporting remote key loading using asymmetric techniques using a “binding” technique, it supports an acceptable method for unbinding in the event of decommissioning.

TB9.25 The tester shall verify the SCRP contains cryptographic keys and functions that can be used to securely provision cryptographic keys and other data to an external system (such as a payment application on a mobile phone). The tester shall detail the relevant key names, function names, and function behavior.

TB9.26 If the device supports IC card acceptance, the tester shall validate that the device supports ECC as described in the guidance.

DTR B10 Encryption Mechanisms

All account data shall be encrypted using only* ANSI X9 *or ISO-approved encryption algorithms (for example, AES, TDES) and should use* ANSI X9 *or ISO-approved modes of operation.

Guidance

All account data shall be encrypted using only ANSI X9 *or ISO-approved encryption algorithms (for example, AES, TDES). The encryption algorithm should use a mode of operation described in* ISO/IEC 10116:2006 *(or equivalent) and follow secure padding guidelines. Any method used to produce encrypted text that relies on "non-standard" modes of operations (for example, format-preserving Feistel-based Encryption Mode (FFX)) shall be approved by at least one independent security evaluation organization (for example, a standards body) and subjected to independent expert review; such methods shall also be implemented following all guidelines of said evaluation and peer review including any recommendations for associated key management.*

The independent expert must be qualified via a combination of education, training, and experience in cryptology to provide objective technical evaluations that are independent of any ties to vendors and special interests. Independent expert qualifications are further defined in the glossary in the PTS POI Modular Security Requirements.

Double-length TDES keys used in connection with SRED can only be used in unique-key-per-transaction implementations as defined in ISO 11568 *for key derivation or transformation—e.g., DUKPT. Double-length TDES keys are not permitted for use in SRED in Master/Session implementations.*

Compliance with B10 is mandatory for any device to be approved against SRED and have SRED listed as functionality provided.

The default configuration of a device approved against SRED must be to encrypt account data unless that data is explicitly excluded through use of a method treated as a sensitive service—i.e., requiring dual control or the use of cryptographic authentication. For example:

- *Where a device implements a "whitelist" function—i.e., the device can be configured to allow for output of some subset of card data in clear text (e.g., for loyalty or other non-PCI cards)—the absence of the whitelist causes all account data to be encrypted. Any whitelists must be cryptographically authenticated by the POI before use or entered manually through the keypad only when the device is in a sensitive state.*
- *Where a device can be configured to enter a state where all account data is not encrypted, the transition to or from this state is treated as a sensitive service.*
- *For devices that allow the enablement (turning on) or the disablement (turning off) of SRED functionality, the enablement must result in the firmware revision number changing and the device providing visual indication of SRED enablement. Disablement must result in the firmware revision number reverting and the device no longer providing visual indication of SRED enablement. The visual indication must not be transient and shall be illustrated by photographic evidence provided in the evaluation report. This is treated as a sensitive function under B5. This must be documented in information provided by the vendor to the entities deploying these devices, including the security policy enumerated in B20.*

(continued)

- *The device has only one operational mode, and the firmware is not able to export the card data—it only provides the data to an authenticated application. The firmware does not allow any other mechanism for card data export.*

In all cases, the device's firmware must manage the cryptographic keys and operations using the device's secure controller (chip), including those for both SRED enablement and SRED relevant protections.

The minimum that must be encrypted, if present, under SRED, is:

- *Full track or equivalent (when aggregated as a single data element), including both Track 1 and Track 2;*
- *Manually entered security validation value—e.g., CVV2, CVC2, and CID2);*
- *Issuer discretionary data (as a single, unparsed field);*
- *Issuer discretionary data – sensitive data (as a parsed field). Any or all portions of the discretionary data are considered to be sensitive unless known to be non-sensitive;*
- *The PAN itself. If the PAN can be parsed, the parts of the PAN in clear text must not exceed the maximum truncation requirements of the associated payment brand when considered in totality with all possible firmware output methods.*

  For example, SRED-compliant POI devices are permitted to output clear-text PAN data to authenticated applications, as well as based on firmware authenticated whitelists. To support acceptable truncation formats for each Payment Brand, a POI may pass the clear-text data to an application for processing, or it may perform truncation and/or encryption of parts of the PAN based on firmware-authenticated whitelists. Whitelists allowing for digits of the PAN to be output in clear text when the system is operating in a format-preserving encryption mode must be considered in addition to any native truncation methods implemented by the POI.

In addition, the following must be encrypted if it is feasible to associate with the corresponding clear-text PAN:

- *Cardholder name*
- *Expiration date*
- *Service code*

For an authenticated application:

- *The application must reside and execute within the physically and logically secure boundary of the target of evaluation.*
- *The application must be cryptographically authenticated by keys secured within at least the level of the Non-PIN Key domain of the POI using algorithms and keys sizes consistent with those stipulated in B10.*

***Note:** This requirement only applies to keys used to encrypt account data where that encryption is used for the purposes of compliance with the external data security requirements of the SRED requirements. For example, keys used to encrypt data only between an encrypting MSR read head and the security processor, where the data is then decrypted and re-encrypted by the security processor with a different, stronger key, are not in scope of this DTR. However, if the TOE is an SCR, it does apply.*

TB10.1 In the event of a non-standard mode of operation being used, the tester shall examine documented credentials of the expert reviewer and assess his/her ability to perform the review. The tester shall also examine documentation supporting the assertion of independence of review and confirm that the reviewer is indeed independent. The tester shall use his or her judgment in determining the appropriate due diligence and explain this.

TB10.2 The tester shall verify that the mechanism used has been implemented following all guidelines of any security evaluation and independent expert review including any recommendations for associated key management.

TB10.3 The tester shall examine all physical and logical test requirements that relate to cryptographic keys used to protect account data and/or other sensitive data and ensure that equivalent protections (including key-management guidance) are applied to cryptographic keys used for account-data protection in accordance with Appendix G.

TB10.4 The tester shall verify that cryptographic keys require compliance with PCI defined criteria for key sizes. Examples of appropriate algorithms and minimum key sizes are stated in Appendix E, along with examples of acceptable hashing algorithms.

DTR B11 Encryption Algorithm Test

The PIN-encryption technique implemented in the device is a technique included in* ISO 9564*.

Guidance

The device must support at least one of the following key-management techniques using AES as described in ANSI X9.24 *and* ISO/IEC 18033-3:

- *DUKPT*
- *Master/Session*

TDES as described in ANSI X9.24 *and* ISO/IEC 18033-3 *may also be supported using the following key-management techniques*:

- *DUKPT*
- *Master/Session*

POI devices shall not support fixed key management for TDES or AES.

If supporting online PIN entry, the device must support ISO PIN-block format 4 and may support any of the following PIN-block formats:

- *ISO format 0*
- *ISO format 1*
- *ISO format 3*

SCRPs shall not support ISO format 1.

SCRPs shall only support ISO format 4 for PIN conveyance from an external system (such as a payment application on a mobile phone which has captured and encrypted the PIN for transferal to the SCRP), using a tokenized version of the PAN provided by the SCRP. This PIN-encryption key shall be unique per transaction.

SCRPs shall perform PIN translation from encryption using a tokenized PAN under the AES key shared with the external system (such as a payment application on a mobile phone) to encryption using the real PAN under a different key shared with the host. This key shared with the host may be either TDES or AES.

For offline PIN:

a) *The PIN that is submitted by the ICC reader to the IC shall be contained in a PIN block conforming to ISO format 2 PIN block. This applies whether the PIN is submitted in clear text or enciphered using an encipherment key of the IC.*

b) *Where the ICC reader is not integrated into the PIN entry device, and PINs are enciphered only for transmission between the PIN entry device and the IC reader, the device shall use one of the PIN-block formats specified in* ISO 9564-1. *Where ISO format 2 PIN blocks are used, a unique-key-per-transaction method in accordance with* ISO 11568 *shall be used. Format 2 shall be used only in connection with either offline PIN verification or PIN-change operations in connection with ICC environments.*

TB11.1 Through source-code review of all PIN entry methods, API guides, and other vendor documentation as appropriate, the tester shall detail all PIN-block formats supported by the POI and reference to relevant documentation. This should include both ISO-compliant PIN blocks, as well as any non-ISO-compliant PIN-block formats. The tester shall note which cryptographic algorithms may be used with each PIN-block format.

TB11.2 From the above list of PIN-block formats, the tester shall confirm that the POI supports ISO PIN-block format 4 if the device supports online PIN. The device may optionally support format 0, 1, or 3 for online PINs, and format 2 for offline PINs if the device supports offline PIN.

TB11.3 The tester shall confirm that SCRPs support the use of a unique-per-transaction, AES PIN-encryption key for conveyance of PINs from an external system (such as a payment application on a mobile phone) and detail the methods used to generate this key.

TB11.4 The tester shall detail all methods that the POI supports for external PIN transfer. This will include encryption of PINs for transport to an acquiring financial institution, as well as transfer to any external card readers or other devices/sub-components outside of the area of the POI validated as secure under Requirement A1.

TB11.5 From the above information, the tester shall confirm that if configured to use ISO PIN-block formats, the POI does not enforce the use of non-ISO-compliant PIN-block formats for any PIN-transfer mechanisms.

TB11.6 For each ISO-compliant PIN-block method supported by the POI, the tester shall perform a PIN entry operation and confirm that:

a) The format used is correct

b) PINs of less than 4 digits are not supported

c) PINs of more than 12 digits are not supported

d) PINs of all values between (and including) 4 to 12 digits are accepted

e) The application does not provide data for padding of format 3 PIN blocks; this padding is instead generated by the POI random number generator. Use source-code review or other information gathered during B7 to validate.

TB11.7 For PIN-block formats 0, 3, and 4, the tester shall confirm whether the PAN values are supplied by the firmware or by an application. If the values are provided by the firmware, the tester shall confirm that the correct digits of the PAN are used in the PIN block. If the values are provided by an application, the tester shall confirm that the POI development documentation details the correct values of the PAN to be used.

***Note:** For SCRPs, this applies to the PIN block outbound from the SCRP to send to the host.*

TB11.8 For SCRPs, the tester shall verify that the SCRP generates a PAN token as stated in B24 for use by an external system (such as a payment application on a mobile phone) for conveyance of an ISO format 4 PIN block to the SCRP.

TB11.9 For SCRPs, the tester shall verify that the SCRP does not support the use of format 1 PIN blocks, and that only ISO format 0, 3, or 4 PIN blocks using the real PAN are supported for translation of the customer PIN as transmitted from an external system (such as a payment application on a mobile phone) for conveyance to a host system.

TB11.10 For SCRPs, the tester shall verify that the SCRP performs PIN translation from encryption using a tokenized PAN under the AES key shared with the external system (such as a payment application on a mobile phone) to encryption using the real PAN under a different key shared with the host. This key shared with the host may be either TDES or AES.

DTR B12 Encryption or Decryption of Arbitrary Data Within the Device

It is not possible to encrypt or decrypt any arbitrary data using any PIN-encrypting key, account-data encryption, data-encrypting key, or key-encrypting key contained in the device.

The device must enforce that PIN encryption, account-data encryption, data-encryption keys and key-encipherment keys have different values.

Guidance

PIN-encryption keys shall only be used to encrypt PIN data. Key-encrypting keys shall only be used to encrypt keys. PIN keys shall never be used to encrypt keys. Key-encrypting keys shall never be used to encrypt PIN data. Data keys shall not be used to encrypt other keys or PIN data.

Account-data encryption keys shall only be used to encrypt account data. Account-data encryption keys shall never be used to encrypt keys.

A secret key used to encrypt a PIN must never be used for any other cryptographic purpose. A key used to protect the PIN-encrypting key must never be used for any other cryptographic purpose.

***Note:** The use of variants is not allowed for AES keys.*

Private keys shall only be used to create digital signatures and to perform decryption operations. Private keys shall never be used to encrypt other keys.

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 intent of the requirement is to help ensure that these keys are not intentionally used for multiple purposes. Thus, the uniqueness check applies for both when the device is initially loaded with these keys and for those that are subsequently loaded. The check must occur within all secret-key hierarchies supported by the device. No two secret keys, regardless of purpose, can have the same value. Keys that are identical except for parity bits must be rejected because they have the same effective value.

This is not intended to require that the device compare keys across different key hierarchies associated with different acquirers.

The evaluating lab may require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.

TB12.1 Through source-code review, the tester shall confirm what methods the POI device uses to confirm the purpose and integrity of each key. The tester shall validate that this does not reduce the effective strength of any key (for example, by including a CRC or purpose tag within the effective value of the cryptographic key).

The tester shall verify the following:

a) PIN-encryption keys are only used to encrypt PIN data.

b) Account-data encryption keys are only used to encrypt account data.

c) Key-encrypting keys are only used to encrypt keys.

d) Data keys shall never be used to encrypt other keys or PIN or account data.

This evaluation activity should be focused at relevant security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB12.2 If the POI supports data-encryption keys, the tester shall confirm what methods are implemented to prevent the use of this function to decrypt PINs. Examples of acceptable solutions are:

- The key-usage information of any downloaded key must be cryptographically bound to the key value using accepted methods, and the device must enforce that the key is only used for the intended use.
- The addition of a new key type (slot) subsequent to the initial configuration of the device causes the zeroization of all other secret keys. Devices supporting remote key-distribution techniques using asymmetric techniques shall only support the use of such techniques for the loading of TMKs. Support shall not exist to use remote key-distribution techniques for working keys (for example, PIN, data, MAC, etc.) unless the key-usage information is cryptographically bound to each individual key.
- Downloaded data keys must not be accepted by the device unless enciphered by a terminal master key that is different than sensitive keys such as the PEK or MAC key types.
- The device does not provide any support for a decrypt data or similar function.

TB12.3 The tester shall verify by testing that the device enforces that data keys, key-encipherment keys, and PIN-encryption keys have different values—for example, by attempting to load keys of different types with effectively the same value. The tester shall attempt to load two keys of the same value into the POI and detail the results. If unsuccessful, the tester shall attempt to load two keys that vary only in the parity bits but produce the same key value. The tester shall detail the results.

DTR B13 Clear-Text Key Security

There is no mechanism in the device that would allow the outputting of a private or secret clear-text key or clear-text PIN, the encryption of a key or PIN under a key that might itself be disclosed, or the transfer of a clear-text key from a component of high security into a component of lesser security.

Guidance

Clear-text secret and private keys and clear-text PINs must not exist in unprotected environments.

The evaluating lab shall require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.

TB13.1 The tester shall examine any documentation—i.e., the Asset Flow Analysis, API programmer's guide, specifications, block diagrams, etc.—containing information relating to the output of clear-text keys and the protection of PINs, the encryption of a key or PIN under a key that might itself be disclosed, and the transfer of a key from a high-security component to a lower-security component

TB13.2 Referencing the table of sensitive-information storage provided in Requirement A4, the tester shall note whether cryptographic keys, customer PINs, or other sensitive data are exported outside the security processor to other components (including memory components) within the POI. The tester shall justify why any such transfer of keys ensures that they remain secure.

TB13.3 The tester shall verify through review of the API guide and source-code review whether the POI allows for injection of customer PINs from an external device. If such functionality is provided by the device, the tester shall detail how this is secured and justify why this ensures that customer PINs and cryptographic keys are not exposed.

TB13.4 Through review of the API guide and operational testing of the POI, the tester shall note whether cryptographic keys are output from the device. Examples of instances where keys may be exported include, but are not limited to:

a) During remote key injection (RKI) key establishment

b) During connection with an external card reader or other secure peripheral

If any key export is allowed by the POI, the tester shall detail the methods used to secure the cryptographic keys during this export and justify why this ensures that customer PINs and cryptographic keys are not exposed.

DTR B14 Transaction Controls

The entry of any other transaction data must be separate from the PIN-entry process, avoiding the accidental display of a cardholder PIN on the device display. If other data and the PIN are entered on the same keypad, the other data entry and the PIN entry shall be clearly separate operations.

Guidance

For devices implementing at the same time a touchscreen for PIN entry and a separate keypad, these must have a default entry mode; switching to the other mode requires an explicit operation on the device, which remains valid for one transaction.

The evaluating lab shall require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.

TB14.1 The tester shall review the source code and detail what protections are provided to ensure that only encrypted data (as verified in Requirement B11) can be output from the POI firmware when the POI is in a PIN entry mode. This evaluation activity should be focused at relevant security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB14.2 The tester shall perform a simulated transaction to verify that the prompt for PIN entry is distinctly separate from all other operations, such as the display of the transaction amount. When prompting for PIN entry, the device must not accept any other data inputs. Control inputs such as “Yes,” “OK,” “Cancel,” or “No” are acceptable.

DTR B15 Logical Management of Display Prompts

All prompts for non-PIN data entry are under the control of the cryptographic unit of the device. If the prompts are stored inside the cryptographic unit, they cannot feasibly be altered without causing the erasure of the unit's cryptographic keys. If the prompts are stored outside the cryptographic unit, cryptographic mechanisms must exist to ensure the authenticity and the proper use of the prompts, and that modification of the prompts or improper use of the prompts is prevented.

Guidance

A8 applies to any components or paths containing clear-text display signals between the cryptographic processor and display unit. B15 applies to devices that allow for updates of prompts or use cryptography to communicate with a display, whether performed by the vendor or the acquirer. C2.4 is appropriate for unattended devices that do not meet any of the aforementioned.

B15 applies to both acquirer and vendor-controlled prompts that are updatable.

Prompts stored inside the cryptographic unit are physically protected according to Requirement A8.

If the device model is to be listed as both an acquirer-controlled and a vendor-controlled display-prompts device, there must be a differentiation so acquirers/merchants can distinguish between the two (for example, different hardware and/or firmware versions).

Controls provide for unique accountability and utilize key sizes appropriate for the algorithm(s) in question. Key-management techniques and other control mechanisms are defined and include appropriate application of the principles of dual control and split knowledge.

A device must automatically record events that are relevant to B15 to a file that is automatically saved. Because each device vendor solution will be unique, the data set that is appropriate to be included in a log file can vary. At a minimum, it is expected that actions that involve cryptographic operations, the user(s), and the time and date of the action will be recorded in the log file. The logs may exist either internally or externally to the device, and a mechanism must be implemented which prohibits the overwriting of log events without proper authentication.

"Non-PIN data" refers to numeric data other than the PIN that is entered via the keypad.

Cryptographic keys that are never used to encrypt or decrypt data or are not used for authentication do not need to be considered secret data and therefore do not need to be erased.

Audio prompts must be considered if applicable.

Dual control must be enforced by an SCD. The SCD can be the PED itself or another device. If an SCD other than the PED enforces dual control, the vendor must either provide the SCD to third parties or describe how an SCD must be used to comply with B15. The description must include an example of a specific, existing SCD that can be purchased and used to comply with B15. The PED must have an API that is compatible with the SCD. The complete solution must be fully developed. It is not acceptable to provide detailed instructions that require users to develop part of the solution.

The controls shall be implemented and enforced by the device. As an exception, a vendor of an unattended device may decide to include into the to-be-approved device scope not only the PIN entry device but also the device controller and the controls implemented to ensure a secure configuration, the device's display management, and properties of the device's cabinet, or procedural controls for the device.

(continued)

For instance, a vendor of an unattended device will use security controls regarding how application programs are written, loaded, and executed on the device. The controls must foresee a suitable level of device management and physical strength for the device's shell and locks, which must be defined by the vendor in accordance with Appendix G.

Authenticity checking provided by TLS is not sufficient to meet B15. TLS is only designed to offer IP security; it does not provide security or integrity of the message interpretive content or context. I.e., it does not prevent message content code (such as HTML5, Java, Javascript, etc.) from requesting an input prompt.

Vendor- or acquirer-certified applications and/or data that are communicated using a managed PKI (in addition to TLS) are acceptable. However, the application and/or data provider must attest that the application and/or data does not contain instructions to make use of a prompt.

TB15.1 The tester shall examine all possible prompts to determine whether any can be used in conjunction with numeric entry in the clear.

TB15.2 The tester shall examine the vendor-supplied documentation to verify that the controls to ensure the authenticity and the proper use of the prompts provide for unique accountability and utilize key sizes appropriate for the algorithm(s) in question. Examples of appropriate algorithms and minimum key sizes are stated in Appendix E, along with examples of acceptable hashing algorithms.

TB15.3 The tester shall note how prompts are stored in the POI and cross-reference this with the information provided in Requirement A4.

TB15.4 The tester shall note how prompts can be updated, specifically including information on if the update mechanism requires changing the hardware, firmware, or some other mechanism (for example, changing the POI application).

TB15.5 Where updates to the prompts require a change to the hardware or firmware, the tester shall confirm from the vendor that this would result in a change to the approved designator of the POI.

TB15.6 Where updates to the prompts require changes to non-firmware items (such as the POI application), the tester shall confirm that any such non-firmware items must be cryptographically authenticated prior to being installed into the POI. The tester shall detail the cryptographic methods used for this authentication, referencing the key table where necessary.

TB15.7 The tester shall note what entities may have access to the keys and mechanisms used to generate the authentication data for such non-firmware items.

TB15.8 The tester shall detail what mechanisms are implemented to ensure that "default" certificates or keys, which may be used for development purposes, are not installed in production devices.

TB15.9 The tester shall detail how the process for generating authentication data (either across firmware, or non-firmware) provides:

a) Dual control

b) Auditability

c) Logging

These control mechanisms must exist and must be suitable even when they are not provided solely by the device.

(continued)

The vendor may alternately provide user documentation detailing the management of cryptographic keys following these principles and implementing the use of a secure cryptographic device for management of these keys. The process exists upstream of the device, but the device must still provide enforcement—for example, validate the MAC or digital signature.

TB15.10 For devices where prompts are acquirer-controlled, the tester shall examine logging documentation provided by the vendor of the actual performance of the techniques and control mechanisms specified by the vendor.

The tester shall examine the vendor-supplied documentation to verify that key-management techniques and other control mechanisms are defined and include appropriate application of the principles of dual control and split knowledge.

TB15.11 The tester shall perform tests to modify the display content and device usage in order to verify that the controls are effective. The tests shall include performing an intended change/update of software and/or display messages and verifying that the result conforms to the specification of the vendor.

DTR B16 Application Separation

If the device supports multiple applications, it must enforce the separation between applications. It must not be possible that one application interferes with or tampers with another application or the OS of the device including, but not limited to, modifying data objects belonging to another application or the OS.

Guidance

As defined in Appendix G, applications, in this context, are functional entities that execute within the secure boundary of the POI and may or may not provide services external to the POI. Applications are typically processes or tasks that execute under the control of an operating system (OS) or software executive routine.

Applications are considered to be separated by access rights. Applications may share data by design.

OS is considered all code, which is responsible to enforce, manage, or change such access rights. Therefore, OS code is necessarily part of the firmware as defined within B1.

The addition of applications that replace or disable the PCI-evaluated firmware functionality invalidates the device approval for each such implementation unless those applications are validated for compliance to PTS POI Security Requirements and listed as such in the approval listings.

Applications must be validated to ensure that:

- *They cannot adversely affect the security features of the product that are relevant to the PCI POI approval.*
- *They cannot modify any of the cryptographic functionality of the POI or introduce new primitive cryptographic functionality. However, new composite functionality that builds on existing primitives is permitted.*
- *The application is authenticated to the POI secure controller by digital signature.*
- *The application can only work on the keys it alone manages and cannot affect or see any other keys. The application must not handle PIN or SRED security-related clear-text keys or key components.*
- *Applications must never process or have access to clear-text PINs or clear-text passwords/authentication codes.*
- *Applications must not share process spaces with each other or with the firmware.*

It is acceptable for applications to be able to initiate changes to security functions, providing the firmware fully implements the authentication mechanisms. When password-based authentication is used, the firmware must manage the password entry and comparison.

Sensitive data in this requirement is defined as:

- *PINs*
- *Passwords/authentication codes*
- *PIN- or SRED-related keys*

Sensitive functions include:

- *Tamper enable/disablement*
- *Removal sensors*
- *Key loading, if dual control is required as per B5*
- *Zeroization*

(continued)

As defined in Appendix G, clear-text PINs and cryptographic key material must be handled in processes which execute firmware code exclusively. It is not acceptable to have firmware code handling sensitive information running in the application context or process.

A mechanism must exist to display the application version upon request.

The vendor must provide clear security guidance for the development and implementation of the aforementioned additional applications. This guidance at a minimum must define procedural controls to ensure that the applications are properly reviewed, tested and authorized. This guidance must be included in the security policy delineated in DTR B20.

Vendors should provide configuration lists and description of the separation mechanism to support answers.

Focal point is for sensitive data and sensitive functions.

TB16.1 The tester shall note whether the POI allows for non-firmware code to be executed. If not, no further testing is necessary for this requirement.

TB16.2 The tester shall analyze the vendor's measures that ensure that the device enforces the separation between applications with security impact from those without security impacts. The tester shall verify that it is not possible that one application interferes with or tampers another application or the OS of the device—especially to access, use, or modify data objects belonging to another application—even if they are distributed over separate components of the device.

TB16.3 The tester shall note whether the POI is designed to allow for non-firmware applications to be executed, and what firmware functions are provided by the processor on which such non-firmware applications would execute (for example, PIN processing, cryptographic-key operations, prompt control, etc.).

TB16.4 **I**f the OS and/or any application with security impact are distributed over separate components of the device, the tester shall verify that the communication in between separated parts is consistent with the separation mechanisms as described by the vendor. The vendor shall provide evidence concerning the communication in between the separated parts and how the communication protocols maintain the separation of applications with security impact from those without security impacts.

TB16.5 The tester shall detail what mechanisms exist within the POI to allow for the execution of non-ROM-based configuration or program data (for example, processors, micro-controllers, FPGAs, etc.). The tester shall note which of these mechanisms execute code that has been considered in-scope (code that impacts these security requirements) of the evaluation, and which do not.

TB16.6 If the POI relies upon the use of different processors to provide for the separation between the firmware and any applications, the tester shall review and briefly describe the method of communications provided between these processors, including any physical interface and API(s).

TB16.7 If the POI allows for different applications to be executed on the same processor, or for the execution of one or more applications on the same processor which is used to execute firmware, the tester shall detail the mechanisms provided to ensure that code and data objects of different applications/firmware are kept separate.

TB16.8 The tester shall review the configuration of the mechanisms described above and confirm that it maintains application separation.

TB16.9 The tester shall note whether the firmware processor(s) provide mechanisms to prevent the execution of memory used to hold data objects. Where such mechanisms exist, the tester shall detail whether they are utilized correctly by the firmware.

TB16.10 If the device allows customers or integrators to install additional applications, the tester shall verify that the POI's design prevents the embedded application from:

- Having access to the top-level master keys that protect the working keys—i.e., the application cannot extract or modify the top-level master key.
- Having access to operator or security-officer functions, and therefore cannot change security configurations or change privileges.

Additionally, the embedded application is separated from the approved device's functionality by an internal security boundary that prevents embedded applications from obtaining any elevated privilege or access to any data belonging to other embedded or host-side applications.

TB16.11 The tester shall verify that clear security guidance for the development and implementation of the aforementioned additional applications is included in the security policy cited in B20. This guidance at a minimum must define procedural controls to ensure that the applications are properly reviewed, tested and authorized.

TB16.12 Using the Asset Flow Analysis, the tester shall describe the path of clear-text PINs from entry through to encryption or transfer to an IC card. Justification must be provided as to why applications cannot access buffers used to store clear-text PINs.

TB16.13 The tester shall describe the software module/s handling PIN- or SRED-related cryptographic keys. Memory used to persistently or temporarily store clear-text keys must be separated from the application.

TB16.14 The tester shall review the API guide and source code provided by the vendor to ensure that no clear-text sensitive data is transferred between firmware and application. Sensitive data includes clear-text PINs, clear-text keys, clear-text key components or clear-text passwords/authentication codes.

TB16.15 The tester shall review the API guide and source code to ensure sensitive functions implement the required authentication controls (as per B5) in firmware and that applications cannot affect or bypass these controls.

TB16.16 The tester shall examine vendor documentation to determine whether the platform is designed to allow applications to execute scripts or code loaded from external sources. If allowed, the tester shall document the function or functions which must be used to parse, verify, and execute/display this script/code.

DTR B16.1 Software Security Domains

If the device supports software with lesser security requirements or which is not developed by the vendor—e.g., applications—it must enforce segregation at least between different software security domains.

Guidance

Software security domains and respective security requirements, in this context, are defined in Appendix G, "Domain-Based Asset Flow Analysis." A vendor may choose to run all software inside a single domain—i.e., the PIN Key domain. In this case, the device is not considered to support applications, and all software is considered firmware, which must comply with all DTR relevant to firmware.

Please refer to Appendix G for a definition of "code" and what is considered software in the sense of this DTR.

If software security domains are supported, the vendor shall perform an asset flow analysis as described in Appendix G It shall at least identify all external interfaces of the device and describe the location and flow of all assets and asset containers in all modes of operation, including power-down and device boot, and for all kinds of transactions.

Vendors shall detail how each domain is mutually segregated from other domains and how the corresponding mechanisms are configured and enforced.

TB16.1.1 The tester shall verify that any domain at any time does not contain any assets, which require a higher attack protection than the rating of the domain.

TB16.1.2 The tester shall analyze the vendor's measures to segregate process spaces. The tester shall verify that process spaces cannot interfere or tamper with process spaces other than their proper children.

TB16.1.3 The tester shall verify the hierarchy of process spaces. Mechanisms to segregate process spaces shall be implemented and configured by process spaces with more access permissions. Such process spaces usually belong to a higher security domain, unless the vendor supports a more granular scheme. They never belong to security domains of lesser attack rating.

TB16.1.4 The tester shall examine any relevant documentation to check whether the asset flow analysis covers all transactions, sensitive services, and operational modes including, but not limited to, key loading, software update, and device boot.

TB16.1.5 The tester shall examine the boot procedure to verify that all loaders, OS layers, etc. are not tagged as a lower domain than the code that they load or maintain.

DTR B16.2 Software Guidance

The vendor must provide clear security guidance consistent with D2 and B4 to all application developers to ensure:

- ***That it is not possible for applications to be influenced by logical anomalies that could result in clear-text data being outputted whilst the terminal is in encrypting mode.***
- ***That account data is not retained any longer, or used more often, than strictly necessary.***
- ***That SRED functions, where provided, are correctly implemented.***

Guidance

As defined in Appendix G, applications are considered to be any code that can be loaded onto the device that is not firmware.

The vendor must provide to the PCI PTS laboratory a guidance document that states the exact scope of the PTS evaluated firmware (down to the level, including version, of libraries and binaries). This shall include all security-relevant APIs to confirm that they are used by the application rather than the application using its own cryptographic primitives and key management.

This document shall be included with the guidance that is available to application developers and must be validated as correct by the PTS laboratory. A reference to this document must be provided in the device security policy that is included with the approval on the PCI website.

Where the POI firmware implements SRED functions, guidance must be provided on how to correctly utilize these functions. This includes application implementations of manual PAN entry where no more than one clear-text PAN digit may be displayed at a time, and a PAN digit displayed during entry must be obfuscated prior to the display of the next digit.

TB16.2.1 The tester shall examine any relevant documentation to ensure that it contains secure programming guidance (consistent with the Secure Software Standard) to assist developers in building secure applications for the device in question.

TB16.2.2 The tester shall examine any relevant documentation to ensure that it contains guidance to assist developers in specifying time limits for how long account data may be retained and often account data should be used before it should be deleted.

TB16.2.3 The tester shall examine any relevant documentation to ensure that it includes guidance for use of the correct API function calls, including sample source code.

TB16.2.4 Where SRED features are implemented, the tester shall examine any relevant documentation to ensure that it includes guidance for the correct use of the SRED APIs. The guidance must also outline that PAN digit entry functions provided by an application cannot display more than one clear-text digit at a time, and that the displayed PAN digit is obfuscated prior to the entry of the next digit.

DTR B17 Minimal Configuration

The operating system of the device must contain only the software (components and services) necessary for the intended operation. The operating system must be configured securely and run with least privilege.

Guidance

The intended operation is considered as the functionality relevant to D2 to prevent any non-firmware application from gaining access to privileged functionality. Least privilege requires that only those components and services that are required to have access to sensitive information, functions, and/or peripherals, be permitted to have such access.

The operating system must be configured to prevent all other components and services from gaining access to the sensitive information, functions, and/or peripherals. Therefore, operating-system code is necessarily part of the firmware as defined within B1.

For the purposes of this requirement, sensitive information, functions, and peripherals, include any method that may provide access to clear-text cryptographic keys and components, customer PINs, passwords/authentication codes used for entry into sensitive states, or other items of information or configuration which is required to meet the logical and physical requirements.

Vendors should provide configuration lists and software specifications to support answers.

If the OS supports multiple privilege levels or user privileges, the least privileges are assigned according to the "need to have" principle; a single supervisory mode implementation—e.g., root or equivalent—is not allowed. An application must have less privilege than the firmware.

TB17.1 The tester shall verify that the security policy enforced by the device does not allow unauthorized or unnecessary functions.

TB17.2 The tester shall verify that API functionality and commands that are not required to support specific functionality are removed whenever possible or disabled if removal is not feasible.

TB17.3 The tester shall examine the methods and tools provided by the vendor that ensure the intended software configuration of the device is maintained, and that updates and release changes do not affect the secure configuration of the OS.

TB17.4 The tester shall determine whether the OS supports multiple privilege levels or user privileges. If so, the tester shall verify that the least privilege is assigned according to the "need to have" principle and that the payment application must have less privilege than the firmware privilege level.

TB17.5 The tester shall note whether the POI implements a commercial operating system, custom operating system, function executive, or other mechanism. If the POI uses a commercial operating system, the tester shall note the name and version of this system.

The tester shall verify that the operating system is enforcing least privilege.

TB17.6 If the POI uses a commercial operating system, the tester shall review publicly available sources of vulnerability information and note whether any vulnerabilities exist for this system. The tester shall note the sources reviewed and any potential vulnerabilities found and justify why any such vulnerabilities are mitigated by the vendor configuration(s).

TB17.7 The tester shall obtain the configuration information for the operating system used in the POI. The tester shall compare this configuration with the supplied documentation and note whether they agree or have differences. If differences are detected, it is necessary to address why these occur with regards to compliance to this requirement.

TB17.8 The tester shall describe the testing and methodology used by the vendor to determine the functions necessary for the POI execution environment. The tester shall justify that this description sufficiently details the steps necessary to reduce the functionality of the POI to only those components and services necessary for the intended operation of the device.

DTR B18 Key Substitution

If the device can hold multiple PIN-encryption keys and if the key to be used to encrypt the PIN can be externally selected, the device prohibits unauthorized key replacement and key misuse.

Guidance

The term "externally selected" means selected by an interface function to the device component that performs the PIN encryption. Both human interfaces and command interfaces are considered, and both direct and indirect.

External selection also includes interference with or manipulation of the data by which the PIN-encrypting device component selects the key to be used.

Keys may be selected through the device keypad, or commands sent from another device such as an electronic cash register. Any commands sent from another device must be cryptographically authenticated to protect against man-in-the-middle and replay attacks.

B18 is not applicable to devices that do not include commands for external key selection or cannot hold multiple key hierarchies related to PIN encryption.

If an application can select keys from multiple key hierarchies, the device must enforce authentication of commands used for external key selection. If the device only allows an application to select keys from a single hierarchy, then command authentication is not required.

TB18.1 Referencing the key table provided in Requirement B9, the tester shall note whether the POI:

a) Provides for a single master key (either symmetric or asymmetric) for all hierarchies into which a PIN key may be loaded, and that this master key is the only key that can be loaded into the POI in clear text.

b) Provides for only one PIN key.

In this case if either of these conditions are true, no further testing is necessary for this requirement.

TB18.2 The tester shall note whether the POI enforces the use of dual control (within the firmware of the POI) to load these keys. The tester shall reference testing of these dual-control features as part of Requirement B5. If dual control is enforced on the POI for the loading of all clear -text keys, then no further testing is necessary for this requirement.

TB18.3 The tester shall note what mechanisms are provided by the POI to authenticate the selection or use of the PIN key and any PIN key-encrypting keys. These mechanisms must be considered a sensitive service and implement dual control or cryptographic mechanisms.

DTR B19 Component Integration Documentation

The vendor must provide adequate documented security guidance for the integration of any secure component into a POI terminal.

TB19.1 For secure component devices the tester shall perform the following tests and describe the results:

a) Visually inspect the secure component as well as examine any additional relevant documentation, such as schematics, housing/frame, data sheets, etc., submitted by the vendor for consistency between documentation and implementation.

b) Verify that procedures exist for the integration documentation to be shipped or otherwise made available to the customer.

c) Verify that the integration documentation is properly maintained—for example, in case of a device update. More specifically, the tester shall examine and document the vendor document-release cycle and assess how it integrates with the device design/manufacturing update process.

DTR B20 Security Policy

A user-available security policy from the vendor addresses the proper use of the POI in a secure fashion, including information on key-management responsibilities, administrative responsibilities, device functionality, identification, and environmental requirements. The security policy must define the roles supported by the POI and indicate the services available for each role in a deterministic tabular format. The POI is capable of performing only its designed functions—i.e., there is no hidden functionality. The only approved functions performed by the POI are those allowed by the policy.

Guidance

The policy must include all configuration settings necessary to meet these security requirements.

If the roles are user-configurable, the security policy must describe the roles that can be configured in the device.

The policy must include procedures for the decommissioning of devices that are removed from service, including the removal of all keying material that could be used to decrypt any sensitive data processed by the device. Procedures may differentiate between temporary and permanent removal.

An English-language version of the security policy must be made available for posting to PCI SSC.

The Security policy should include the following sections:

- *General Description (DTRs B20.1 – B20.6)*
- *Installation and Guidance (DTRs B20.7 – B20.16)*
- *Operation and Maintenance (DTRs B20.17 – B20.24)*
- *Security (DTRs B20.25 – B20.33)*

This is the minimum information that must be presented. Additional information may be presented.

See Appendix I for an example of an acceptable Security Policy layout.

TB20.1 The tester shall examine the security policy to verify that the device is properly identified. Model name, hardware version, and software version information should be included in the identification of the approved device. The tester shall validate that the security policy includes actual (not computer-generated or similar) pictures of the device, and how to validate the hardware, firmware, and application versions and the exact approval class and use case of the device including the specific POI Security Requirements version validated and approved against. The tester shall check and confirm that the Security Policy is well-formatted, accurate, consistent, complete, and does not contain ambiguous or misleading information. The tester shall verify any URL in the security policy as being valid and usable.

TB20.2 Confirm that the security policy clearly outlines the approval class and the method of use for which the device has been approved—e.g., handheld and/or desktop—and that the security policy clearly notes that use of the device in an unapproved method will violate the PCI PTS approval of the device.

TB20.3 The tester shall confirm that the security policy contains references to any software-development guidance required for compliance, if applicable, with the Open Protocol and SRED-relevant requirements. This documentation must clearly outline which functions, APIs, or modes of operation of cryptographic functions (such as cipher suites) have been evaluated by the PTS laboratory for securing cardholder data.

TB20.4 The tester shall confirm that the security policy includes any communication methods, including wireless, and any protocols, including security protocols used by the device. Use of any method not listed in the policy invalidates the device approval.

TB20.5 The tester shall confirm that if the device supports SSL, the security policy must clearly state that it is inherently weak and should be removed unless required on an interim basis to facilitate interoperability as part of a migration plan.

TB20.6 The tester shall confirm the security policy defines and documents all hardware and firmware version options, both security and non-security relevant. Security-relevant options must be exhaustively defined and documented. For non-security options, the security policy must include a description of the option, but not a full list of all possible values.

TB20.7 The tester shall examine the security policy to verify that it identifies all conditions (for example, voltage, humidity, and temperature) that will cause environmental failure-protection mechanisms to trigger.

TB20.8 The tester shall confirm that the security policy includes all configuration settings necessary to meet the security requirements defined in this document.

TB20.9 The tester shall confirm that the security policy contains specific details on how to change any default values, including passwords/authentication codes and certificates.

TB20.10 The tester shall confirm that the security policy contains any installation or user guidance as required by the laboratory for compliance with the PCI PTS requirements. Examples of such required guidance, includes but is not limited to:

- Outlining methods to check the slot of the customer ICC acceptor for shim devices, as required by DTR A13, including a diagram of the card slot illustrating the process; or
- Providing instructions with regard to tamper evidence, overlay detection, inspection frequency, and procedures.

TB20.11 The tester shall examine the security policy and relevant vendor documentation to verify that documentation includes procedures for authentication of the device when received via shipping. Note that this may include visual or cryptographic methods

TB20.12 The tester shall confirm that the security policy defines guidance for the proper implementation of any Open Protocols that are part of the approval.

TB20.13 For PIN entry devices that appear as a single device—i.e., where two physically and electronically distinct devices—e.g., a PED and a commercial off-the-shelf (COTS) device such as a mobile phone—appear as a single device through the use of the plastics to mask the connectivity—the tester shall confirm that the security policy states how the device is compliant to PCI PTS POI Evaluation Technical FAQ #26.

TB20.14 The tester shall confirm that for any handheld PIN entry device, if the device does not support SRED encryption, the security policy must clearly state that the system cannot be implemented to connect to a tablet or mobile phone, and any such use will violate the approval of the device.

TB20.15 The tester shall confirm that for any device having beacons, the security policy states how those are compliant to PCI PTS POI Evaluation Technical FAQ #14.

TB20.16 For unattended payment terminal designs where the ICCR slot cannot be positioned straight (horizontal) to the cardholder, the tester shall confirm that the security policy stipulates the allowed installation height ensuring a sufficient view on the card slot entry area as defined in PTS POI Evaluation DTR A14.

TB20.17 The tester shall confirm that for devices that allow the enablement (turning on) or the disablement (turning off) of SRED functionality, the security policy documents how this is in accordance with PTS POI Evaluation DTR B23.

TB20.18 The tester shall confirm that the security policy includes procedures for the decommissioning of devices that are removed from service, including the removal of all keying material that could be used to decrypt any sensitive data processed by the device. The procedures may differentiate between temporary and permanent removal.

TB20.19 For privacy shielding the tester shall perform the following tests:

a) If the POI has been approved for use with a privacy shield, the tester shall confirm that the security policy provides a picture of the approved privacy shield as properly installed and tested by the lab.

b) If the POI has been approved without a privacy shield, the tester shall confirm that any guidance on the use of the POI device as required by Appendix A.2 is provided in the security policy document.

TB20.20 The tester shall confirm that the security policy contains information on all ways the device will indicate a tamper-response event, and any requirements for the return of this device to the vendor for examination following such an event (as required for compliance to DTR A1). This shall include an illustration to show the user an actual tamper-response display message and accurate information on any other tamper-responsive behaviors.

TB20.21 The tester shall examine the security policy and relevant vendor documentation to verify that any periodic maintenance procedures required for the secure operation of the device are included in the security policy.

TB20.22 The tester shall examine the security policy to verify that it identifies all self-tests that the module performs, procedures that an operator may need to initiate any self-tests, and the conditions under which each self-test is executed (for example, power up, periodic, conditional, on operator demand).

TB20.23 The tester shall examine the security policy to verify that it identifies all roles supported by the device and indicates the services and permissions available for each role.

TB20.24 The tester shall examine the security policy and relevant vendor documentation to verify that the device has update and patch procedures required for the secure operation of the device and that the procedures are included in the security policy. The policy will include both local and remote update and patch downloading procedures for software, firmware, and configuration parameters, including those necessary for meeting the Open Protocols requirements.

TB20.25 The tester shall confirm that the security policy defines how the device is compliant to PCI PTS POI Evaluation DTR B1, for any device where the required memory re-initialization (security) cycle last longer than 24 hours. Specifically, how the firmware of the device during the cycles' adjustment processes does not allow any security cycle to last longer than the combined maximum durations of the security cycle and the business cycle (48 hours).

TB20.26 The tester shall confirm that the security policy contains specific details on the cryptographic algorithms (TDES, SHA-2, etc.) and key-management methodologies supported by the device. It is not required that this be expressed as a key table as required by DTR B9, but it must detail the specific keys and usages of these keys for all key-management methods exposed to the device operators. Key-management operations that are only used within the device, or between integrated device components, are not required to be detailed.

TB20.27 The tester shall confirm the security policy of the POI and confirm that it clearly outlines the exact details of the acquirer/processor key-management systems supported by the POI—i.e., simply using the term "MK/SK" is not sufficient—and specifies that use of the POI with different key-management systems will invalidate any PCI approval of this POI. E.g., Key name, purpose/usage, algorithm, key size, form factor in which the key is loaded to the device, number of keys of each usage type supported.

TB20.28 The tester shall confirm that the security policy contains specific details on any account-data protection schemes employed—e.g., algorithms used, format-preserving encryption techniques—and whether the device supports the pass-through of clear-text account data using techniques such as whitelisting.

TB20.29 The tester shall confirm that the security policy contains specific details on how key loading must be performed for operation of the device. This must include any requirements for dual control and split knowledge, as required by DTR B5 and assessed by the PTS laboratory. The security policy must categorize the key-loading techniques supported as either:

- Clear-text key components through the keypad (TB5.5 a),
- Clear-text key injection (as per TB5.5 b),
- Symmetric encrypted keys (TB5.5 c), remote asymmetric key loading (TB5.5 d), or
- Local asymmetric key loading (similar to TB5.5 d, except the local environment provides the mutual authentication, not cryptography).

TB20.30 The tester shall confirm that the security policy contains specific details on how any signing mechanisms must be implemented. This must include any "turnkey" systems required for compliance with B15, or any mechanisms used for authenticating application code as assessed under Requirements B2.1.

TB20.31 The tester shall examine the security policy to verify that it states that keys should be replaced with new keys whenever the compromise of the original key is known or suspected, and whenever the time deemed feasible to determine the key by exhaustive attack elapses, as defined in *NIST SP 800-57-1.*

TB20.32 The tester shall examine the security policy to verify that it describes any PCI-approved components (as defined in the Device Testing and Approval Program Guide) that are used and how they are used in conjunction with this device.

TB20.33 The tester shall confirm that the security policy defines how the device is compliant to PCI PTS POI Evaluation DTR D12 for any device implementing BLE. Specifically, implementations must use version 4.2 or higher and use LE Security Mode 1 Level 4 (Secure Connections) only. Just Works cannot be used at any time. The device must not support or allow for the use of insecure communication options such as, but not limited to, LE Security Mode 2, and levels 1, 2, and 3 of LE Security Mode 1 and the "Just Works" secure pairing option of Security Mode 1.

DTR B21 PIN Protection During Transmission Between Device and ICC Reader

If the PIN entry device and the ICC reader are not integrated into the same secure module, and the cardholder verification method—i.e., required by the IC card—is determined to be:

- ***An enciphered PIN, the PIN block shall be enciphered between the device encrypting the PIN and the ICC reader either using an authenticated encipherment key of the IC card or in accordance with ISO 9564.***
- ***A clear-text PIN, the PIN block shall be enciphered from the device encrypting the PIN to the ICC reader (the ICC reader will then decipher the PIN for transmission in clear-text to the IC card) in accordance with* ISO 9564*.***

If the PIN entry device and the ICC reader are integrated into the same secure module, and the cardholder verification method is determined to be:

- ***An enciphered PIN, the PIN block shall be enciphered using an authenticated encipherment key of the IC card.***
- ***A clear-text PIN, encipherment is not required if the PIN block is transmitted wholly through a protected environment (as defined in* ISO 9564*). If the clear-text PIN is transmitted to the ICC reader through an unprotected environment, the PIN block shall be enciphered in accordance with ISO 9564.***

Guidance

DTRs A1 and A13 verify the physical protections relevant to DTR B21.

For an SCRP, this requirement is applied without consideration of the conveyance of the PIN from an external system (such as a payment application on a mobile phone). The conveyance of the PIN from any such external system to the SCRP shall always use ISO format 4 for that conveyance.

SCRPs with ICCRs shall support both enciphered and clear-text PIN for offline PIN authentication.

B21 requires that the following be met:

- *A clear-text PIN from the PIN entry device to the ICC reader is never permitted except when the PIN entry device and ICC reader are integrated into the same secure module.*
- *When the cardholder verification method is determined to be an enciphered PIN, the encipherment must occur within the PIN entry device itself or a secure component of the device. The PIN must be enciphered in accordance with* ISO 9564 *for secure transport between the PIN entry device and the secure component. The PIN must be encrypted immediately after PIN entry is complete and has been signified as such by the cardholder for example, via pressing the enter button.*
- *In order to receive an approval for offline PIN entry, a device must be capable of supporting both clear-text and enciphered PINs.*
- *The authentication must occur in a secure component of the device, such as the PIN pad or ICCR. This includes the authentication of the ICC public key(s) as well as the associated issuer public key in the certificate chain, up to the applicable payment-brand key.*
- *The evaluating lab may require copies of source code and assistance from the vendor to make a systematic review of relevant security functions.*
- *PINs enciphered only for transmission between the PIN entry device and the IC reader must use one of the PIN-block formats specified in* ISO 9564*. Where ISO format 2 is used, a unique-key-per-transaction method in accordance with* ISO 11568 *shall be used.*

TB21.1 The tester shall note whether the customer ICC acceptor is integrated into a single, secure enclosure with the customer PIN entry mechanism and security processor, or the ICC acceptor is a separate part.

TB21.2 The tester shall confirm from the vendor documentation that the POI supports both clear-text and encrypted offline PIN. The tester shall detail the APIs that are used for these purposes and justify how these ensure that the customer PIN is always encrypted by the firmware of the POI and is never passed to the application in clear text.

TB21.3 If the ICC acceptor is a separate part, the tester shall detail how the customer PIN is encrypted for transmission between the two parts. This encryption must use *ISO 9564*-compliant PIN blocks and encryption mechanisms (an authenticated encipherment key of the ICC, TDES, or AES). The tester shall review the source code that performs the PIN-block formatting and encryption and confirm it is compliant to *ISO 9564.* This evaluation activity should be focused at relevant security-critical sections of the source code to provide an optimum balance between use of evaluation resources against evaluation goals overall.

TB21.4 If the ICC acceptor is a separate part, the tester shall detail what mechanisms are implemented to ensure that the key used to encrypt customer PINs is not used for any other purpose (for example, this key is not used to encrypt all data between the two parts).

TB21.5 If the ICC acceptor is a separate part, the tester shall perform at least two transactions using the same customer PIN and card details and note whether the encrypted PIN block transferred between the two parts is the same.

TB21.6 If the ICC acceptor is a separate part, the tester shall detail what mechanisms are implemented to ensure that PIN data transmitted between the EPP and card reader cannot be “replayed” during, or as part of, any subsequent transaction.

TB21.7 The tester shall perform at least two test transactions with a card that requires an encrypted PIN. The tester shall capture the data sent to the customer card in each case and note whether the two encrypted PIN blocks have the same value.

TB21.8 The tester shall perform at least two test transactions with a card that requires a clear-text PIN. The tester shall capture the data sent to the customer card in each case and note whether the PIN is correctly formatted as an ISO 9564 format 2 PIN block.

TB21.9 The tester shall detail how the POI authenticates the customer ICC PIN-encryption public key. If this authentication is performed by the application, the tester shall confirm that the application authentication mechanisms have been tested as part of Requirement B2.1.

TB21.10 The tester shall perform four offline transactions and monitor the value of the unpredictable number (UN) provided by the terminal in each case. Through observation of this value, the tester shall confirm that the unpredictable number is not implemented as a simple counter or some other easily guessable value. The tester shall also review the source code used to generate the UN and confirm that the terminal uses the random number generator of the terminal, evaluated under Requirement B7, to generate the unpredictable number values.

DTR B22 Remote Access

If the device can be accessed remotely for the purposes of administration, all access attempts must be cryptographically authenticated. If the authenticity of the access request cannot be confirmed, the access request is denied.

Guidance

The authentication must not be performed by a component of lesser protection strength than the one for which the access is intended, OR the authentication must be performed by the target component.

B2.1 and B2 address application loads and firmware, application, and configuration updates. B22 is intended to address other types of administration activities, such as those more prevalent with unattended devices. In any case, unless there is not any impact (for example, the load itself is cryptographically authenticated at the target), a secure session should be established (for example, TLS) for those communications.

TB22.1 The tester shall verify that the device cryptographically authenticates remote access attempts. This will be accomplished, for example, by performing a simulated remote access attempt.

TB22.2 The tester shall verify that the device rejects unauthorized remote access attempts. This will be accomplished, for example, by performing a simulated remote access update with inadequate or modified authentication information.

TB22.3 The tester shall determine by which component the authentication is performed.

TB22.4 The tester shall examine the vendor-supplied documentation to verify that the controls provide for unique accountability and utilize key sizes appropriate for the algorithm(s) in question.

TB22.5 The tester shall determine the level of protection for the external component involved in firmware/software updates and that the authentication of software updates is performed by a component of equal or greater strength.

TB22.6 The tester shall examine any relevant documentation—such as schematics, data sheets, vendor test procedures, and test reports, etc.—submitted by the vendor to ensure freshness of messages exchanged in remote access attempts.

If clock-based mechanisms are used to ensure freshness, the tester shall verify that the device is capable of synchronizing with a secure time source. If randomly generated nonces are used, the tester shall verify that nonces are chosen from a set sufficiently large to mean that the probability of the same nonce being used twice is effectively zero.

DTR B23 Output of Clear-text Account Data

When operating in encrypting mode, there is no mechanism in the device that would allow the outputting of clear-text account data except as described in DTR B23. Changing between encrypting and non-encrypting modes of operation requires explicit authentication.

Guidance

PAN data that is encrypted, hashed (with salt), masked or truncated PANs may be outputted from the device. Truncated PANs are typically defined as a maximum of the first six and the last four digits. Individual card brands have defined guidance for truncation that varies for individual cards based on PAN length, IIN/BIN type, etc. If the implementation permits truncation methods that don't conform to the brand guidance, a determination must be made if the truncated digits offer sufficient protection against attacks designed to predict valid, full PANs (with longer BIN ranges). This would partially depend on the potential universe of PANs that could be included; and if the vendor is required to output more PAN data than permitted by the card brands, it must demonstrate that the probability of PAN recovery is equivalent to that which is recommended by the card brands. If using truncation, any removed segment cannot be replaced with a hashed version of any component of the original PAN. Truncated and hashed versions of the same PAN must not be transmitted together unless encrypted.

Encrypting mode is defined to be when the device's encryption of account-data functionality is enabled and operational. Changing between modes is considered a sensitive service as stated in B5 and B6 and therefore requires that authentication use dual-control techniques when entering sensitive information through a secure user interface, or cryptographic techniques when entering electronic data.

If whitelist(s) are utilized to exclude card data from mandatory encryption, the whitelist shall be cryptographically authenticated either prior to being instantiated in the device or before being utilized.

For devices that allow the enablement (turning on) or the disablement (turning off) of SRED functionality, the enablement must result in the firmware revision number changing and the device providing visual indication of SRED enablement. Disablement must result in the firmware revision number reverting and the device no longer providing visual indication of SRED enablement. The visual indication must not be transient. This must be documented in information provided by the vendor to the entities deploying these devices.

A secure card reader intended for use with a non-PTS-approved device such as, but not limited to, a mobile phone or tablet; or an SCRP, is only allowed one state, and that is to encrypt all account data. It cannot be configured to enter a state where account data is not encrypted.

TB23.1 The tester shall examine any log or trace files generated by the device to determine whether they support the assertions made by the vendor.

TB23.2 The tester shall verify from vendor documentation that sensitive services are entered, used, and exited securely and that mode transitions do not reveal or otherwise affect sensitive information.

TB23.3 The tester shall verify from vendor documentation and from functional testing that changing modes of operation requires authentication using dual-control techniques when entering sensitive information through a secure user interface, or cryptographic techniques when entering electronic data.

TB23.4 If access to changing modes of operation requires input by the keypad, the tester shall verify that the protections, such as the following, are also afforded to data entered while accessing sensitive services:

- Data inputs cannot be discerned from any displayed characters.
- Data inputs cannot be discerned by monitoring audible or electro-magnetic emissions.
- Sensitive data is cleared from internal buffers upon exiting the sensitive mode.

The testing shall include:

- Entering data while changing modes.
- Document review.

TB23.5 If mode transitions require input by a separate interface device, such as a key loader, the tester shall document the mechanism(s) and methodology used.

TB23.6 If the device allows for a change of modes between encrypting and non-encrypting, the tester shall verify that:

- The authorization implements the principles of dual control.
- Sensitive information required for the authorization (for example, passwords/authentication codes or cryptographic mechanism) is initialized or used in a way, that prevents replay at the same or a different device.
- An authorized switch must provide traceability and accountability.
- If changes to the mode of operation can occur remotely, the tester shall verify that a cryptographic authorization mechanism consistent to that used for remote access authentication (per B22) is implemented.

TB23.7 Where the firmware supports the output of truncated PAN values, the tester shall confirm that when operating in encrypting mode, the POI enforces truncation that meets relevant brand requirements and PCI FAQs. Examples of relevant FAQs includes, but may not be limited to, PCI DSS FAQ 1091.

DTR B23.1 Protection of Clear-text Data

When operating in encrypting mode, the secure controller can only release clear-text account data to authenticated applications executing within the device.

Guidance

The device shall only release clear-text account data to authenticated applications. The device must never release any other clear-text data such as cryptographic keys.

TB23.1.1 The tester shall verify that the vendor has identified all data that is provided to authenticated applications.

DTR B24 Surrogate PAN Values

If the device is capable of generating surrogate PAN values to be outputted outside of the device, it is not possible to determine the original PAN knowing only the surrogate value. Where a hash function is used to generate surrogate PAN values:

- ***Input to the hash function must use a salt with minimum length of 64 bits.***
- ***The salt is kept secret and appropriately protected.***
- ***Disclosure of the salt cannot occur without requiring an attack potential of at least 16 per device for identification and initial exploitation with a minimum of 8 for initial exploitation, as defined in Appendix B.***

Guidance

To support ancillary business process, surrogate PAN values may be generated by the device. The probability of determining the original PAN knowing only the surrogate value should be no better than a random guess.

SCRPs must tokenize the PAN data to send to any external system (such as a payment application on a mobile phone) for use in formatting the ISO format 4 PIN block. An example of an acceptable PAN token is one described in ANSI X9.119-2 *using a format-preserving scheme.*

A cryptographic salt comprises random bits that can be input into a cryptographic function. Random bits shall be generated such that probability of the same random bits being output is statistically insignificant. A known salt value may compromise the effectiveness of the cryptographic function.

The salt may be unique per transaction, unique per a group of transactions, unique per device, or unique per merchant.

- *Salts that are unique per transaction or otherwise unique per device must be generated by the transaction device.*
- *Salts that are unique per merchant are generated outside the transaction device and require loading to each merchant device. The vendor must supply documentation to the merchant/acquirer processor on how to securely load the salt values and that this loading is treated as a sensitive service in accordance with B5.*

TB24.1 If a cryptographic key or algorithm is used to generate surrogate values, the tester shall examine the vendor-supplied documentation to verify that the controls utilize algorithms and/or key sizes appropriate for the surrogate creation mechanism in question.

TB24.2 Where a hash function is used to generate surrogate PAN values:

a) The tester shall verify test information provided by the vendor to assess whether the random numbers are sufficiently unpredictable. The tester shall use a suitable test method consistent with B7.

b) The tester shall verify by testing that the device generates salt values with a minimum length of 64 bits.

c) The tester shall develop attack scenarios to defeat or circumvent the protection mechanisms dealing with salt data and functions that interact with salt data, using attack scenarios. The tester shall detail whether steps are based on actual testing or on assumptions and provide justification for any use of assumptions rather than actual testing.

The tester is not required to perform the attack entirely but may perform all or part of the attack to verify its validity. The calculation shall be based on the scheme depicted in Appendix B. If an attack scenario can be developed that yields an attack potential of less than 16 per device for identification and initial exploitation or less than 8 per device for initial exploitation only, as defined in Appendix B, the vendor assertion cannot be verified. At least one attack scenario shall be presented, in a format consistent with the examples shown in Appendix B in this document.

The tester shall present evidence of the test methodologies followed and the validation results.

DTR B25 Exhaustive PAN Determination

The device has characteristics that prevent or significantly deter the use of the device for exhaustive PAN determination.

Guidance

The following are examples of techniques that may be used to prevent an exhaustive PAN determination attack, such as one producing random transactions through the device until the ciphertext produced equals the ciphertext recorded when the device was in operational use:

- *Use of a unique-key-per-transaction technique. (Prevents the attack.)*
- *Limiting the rate at which the device will encrypt PANs. (Deters the attack.) For example, the function would be a maximum of the throughput that could be achieved through the physical interface during intended usage.*

TB25.1 The tester shall perform functional testing to verify the device characteristics regarding B25.

DTR B26 Secure Enablement Tokens

Secure enablement tokens are required from the attestation and monitoring system for the SCRP to accept and/or process payments.

Guidance

The SCRP must use either asymmetric or symmetric techniques to authenticate the enablement token. Any private or symmetric keys present on the SCRP for this purpose must be unique per device.

The SCRP may continue to provide functions necessary for the resumption of payment acceptance and processing by the SCRP, even when an enablement token is not provided within the acceptable time period. Examples of functions permitted in the absence of an enablement token may include the ability to establish a secure channel with the external system interfacing to the SCRP (such as a payment application on a mobile phone), providing data required for the attestation of the SCRP device, and the supply of random data for the purposes of seeding external DRNGs.

TB26.1 The tester shall confirm that without a cryptographically secure enablement token being supplied to the SCRP, the SCRP will cease accepting payment cards within a time no more than 24 hours. Re-enablement of the SCRP to accept payment cards may be performed at any time with the validation of a fresh enablement token.

TB26.2 The tester shall confirm that an enablement token is required to be provided to the SCRP upon power-up, prior to the SCRP accepting payment cards, except in the case of offline (store-and-forward) payment. The token may be replayed as long as not expired.

TB26.3 The tester shall document the features of the enablement token that prevent replay of this value. The tester shall attempt to replay the enablement token and confirm that this does not allow for the continued operation, or re-enablement, of the SCRP.

DTR Module 2: POS Terminal Integration Requirements

C – POS Terminal Integration Derived Test Requirements

DTR C1.1 Integration of PIN Entry Functions

The logical and physical integration of a PCI-approved secure component (or components) into a PIN entry POI terminal must not impact the overall PIN protection level.

Guidance

A PIN-handling component of the device (for example, the EPP) never outputs information to another component (for example, a display or a device controller) allowing the differentiation of the PIN digits entered.

DTRs C1.1 to C1.2 specifically care about the PIN entry function and how it is integrated into a payment terminal (irrespective of the form factor), whereas DTRs C2.1 to C2.5 aim at ensuring that components are being properly integrated. As such, DTRs C2.1 to C2.5 target compound architectures and are not relevant to, for example, countertop devices.

TC1.1.1 The tester shall examine that the integration of every PCI-approved secure component has been performed strictly according to the component manufacturer recommendations.

TC1.1.2 The tester shall verify that the failure, removal, or absence of a PCI-approved secure component does not lead to another approved secure component revealing any PIN-related sensitive information and does not lead the PIN entry POI terminal to fall back into a non-safe operating mode.

TC1.1.3 If a secure component directly controls the display, the tester shall verify that it does not display any digits of the PIN value by performing a transaction where a PIN number is entered.

TC1.1.4 If the secure component supports a signaling mechanism at the interface to allow an external display to echo key presses, the tester shall verify that it does not display any digits of the PIN value by performing a transaction where a PIN number is entered.

DTR C1.2 Overlay Attack Protection

The PIN pad (PIN entry area) and the surrounding area must be designed and engineered in such a way that the complete device does not facilitate the fraudulent placement of an overlay over the PIN pad.

An overlay attack must require an attack potential of at least 18 for identification and initial exploitation, with a minimum of 9 for initial exploitation, as defined in Appendix B.

Guidance

> *Recessed keypads should be avoided when possible, as it increases the risk of successfully implanting an overlay and yet remain unnoticed for an extended period of time. If recessed area(s) are used, other mechanisms must be implemented to deter overlay placement.*
>
> *Complete covering of the device is not in scope; only the keypad area is considered.*
>
> *If the PIN pad implements a PIN shield, attack scenarios must take this into account.*
>
> *Successful overlay attack can show moderate differences in shape and color from the original device, and yet remain unnoticed by typical cardholders. No expectation shall be made upon cardholder's ability to distinguish between legitimate devices and overlaid ones.*

TC1.2.1 The tester shall develop attack scenario(s) for the fraudulent placement of an overlay over the PIN pad. If an attack scenario can be developed that yields an attack potential of less than 18 per device for identification and initial exploitation or less than 9 per device for initial exploitation only, as defined in Appendix B, the vendor assertion cannot be verified. At least one attack scenario shall be presented, in a format consistent with the examples shown in Appendix B in this document.

The tester may perform any test needed to validate the attack scenario. The tester shall detail whether steps are based on actual testing or on assumptions and provide justification for any use of assumptions rather than actual testing. Calculations shall include evidence justifying particular rating levels as being appropriate.

The tester shall present evidence of the test methodologies followed and the validation results.

DTR C2.1 Integration Vulnerabilities

The logical and physical integration of an approved secure component into a PIN entry POI terminal does not create new attack paths to the PIN.

Guidance

Any secure component integrated into a PIN entry POI terminal submitted for evaluation has a clearly identified physical and logical security perimeter (related to PIN entry and card-reading functions).

Logical integration encompasses communication protocols between secure devices and non-secure devices. Devices pairing, unbinding, mutual authentication, and other security-sensitive mechanisms enter this category. Special care is needed when combining devices that were approved in isolation.

Physical integration of components encompasses, for example, integration into a cabinet.

Protocol fault injections are covered by Requirement D1, whereas protocol abuses through regular requests are covered by this requirement.

DTRs C1.1 to C1.2 specifically care about the PIN entry function and how it is integrated into a payment terminal (irrespective of the form factor), whereas DTRs C2.1 to C2.5 aim at ensuring that components are being properly integrated. As such, DTRs C2.1 to C2.5 target compound architectures and are not relevant to, for example, countertop devices.

TC2.1.1 The tester shall examine that the integration of the approved secure component into the PIN entry POI terminal has been performed strictly according to the component manufacturer's recommendations.

TC2.1.2 The tester shall verify that the failure of secure component does not lead the PIN entry POI terminal to fall back in a non-safe mode—i.e., no more protecting the PIN as per requirements.

TC2.1.3 The tester shall verify that the protocols used for secure communication between secure components are strong enough to deter attacks aiming at stealing sensitive information, including those involving impersonation of one component, and replay attacks.

DTR C2.2 Protection Against Card Trapping

The PIN entry POI terminal is equipped with mechanisms to prevent attacks aiming at retaining and stealing the payment card—e.g., Lebanese Loop attack.

Guidance

Implementations may consist of shutters, detection of any obstruction in the card path, or the like. Alternatively, the protection may be implemented via software and leveraging detection mechanisms already fitted in equipment.

TC2.2.1 The tester shall visually inspect the device's card reader and/or the device in general to verify the assertions provided by the vendor.

TC2.2.2 The tester shall perform suitable test to verify the functionality of the mechanisms.

DTR C2.3 PIN Entry Interface Segregation

There is a clear logical and/or physical segregation between secure components and non-secure components integrated into the same device.

Guidance

More specifically, there is a clear logical and/or physical segregation between PIN security-related interfaces and generic cardholder POS interface (keys, display, card-reading, etc.).

Implementation must unambiguously lead the cardholder to use the interfaces assigned for secure PIN entry functions from other generic cardholder POS interfaces

Distinct housing materials or colors, etc. may constitute such physical segregation.

The usage of visually separate displays (or a single display with clearly separated entry interfaces) may support the logical separation of interfaces (or security functions).

TC2.3.1 The tester shall visually inspect the PIN entry device as integrated inside the POS Terminal to verify the assertions provided by the vendor.

TC2.3.2 The tester shall establish a comprehensive list of all components, together with their relationship and qualification against security, and verify that they are physically or logically segregated.

TC2.3.3 If the vendor does not provide with the final PIN-enabled payment application, the tester shall verify that the vendor provides third party developers with the appropriate documentation on how to implement the transaction flow, such as it is reasonably obvious for a cardholder to distinguish whether or not he or she is about to enter his or her PIN on the device.

DTR C2.4 User Interface Consistency

The POI (application) must enforce the correspondence between the display messages visible to the cardholder and the operating state—i.e., secure or non-secure mode—of the PIN entry device—e.g., by using cryptographic authentication.

If commands impacting the correspondence between the display messages and the operating state of the PIN entry device are received from an external device—e.g., a store controller—the commands enabling data entry must be authenticated.

The alteration of the correspondence between the display messages visible to the cardholder and the operating state of the PIN entry device cannot occur without requiring an attack potential of at least 18 per POI for identification and initial exploitation with a minimum of 9 for initial exploitation, as defined in Appendix B.

Guidance

A8 applies to any components or paths containing clear-text display signals between the cryptographic processor and display unit. B15 applies to devices that allow for updates of prompts or use cryptography to communicate with a display, whether performed by the vendor or the acquirer. C2.4 is appropriate for unattended devices that do not meet any of the aforementioned.

"Non-PIN data" refers to numeric data other than the PIN that is entered via the keypad.

The ability to physically access the connection between devices (for example, EPP, UPT controller, and ICC reader) must not facilitate attacks to interfere with that correspondence and especially to collect clear PIN data (for example, commands are authenticated and/or enciphered). Keys used for such protocols must only be used for this purpose and may then be stored and used in the UPT controller in the clear. It is not required to use a cryptographic module in the controller for that purpose.

Audio prompts must be considered if applicable.

TC2.4.1 The tester shall detail where prompts used for non-PIN entry are stored within the POI and describe the protections implemented to protect these prompts. The tester shall reference this information to the table of sensitive information provided in DTR A4.

TC2.4.2 The tester shall describe the path from the display to the processing element that controls the display. The tester shall verify, by testing, that the ability to physically access the connection between devices does must not facilitate attacks to interfere with that correspondence. If commands are authenticated and/or enciphered, the existence and efficiency of this mechanism must be verified.

TC2.4.3 The tester shall examine the vendor-supplied documentation to verify that the controls provide for unique accountability and utilize key sizes appropriate for the algorithm(s) in question. Examples of appropriate algorithms and minimum key sizes are stated in Appendix E, along with examples of acceptable hashing algorithms.

TC2.4.4 If commands impacting the correspondence between the display messages and the operating state of the device are received from an external device (for example, a store controller), the tester must verify that the commands enabling data entry are authenticated and that the mechanisms are efficient.

TC2.4.5 The tester shall perform tests to modify the display content and device usage in order to verify that the controls are effective. The tests shall include performing an intended change/update of software and/or display messages and verifying that the result conforms to the specification of the vendor.

TC2.4.6 The tester shall develop attack scenarios to circumvent the control of the display by the device. If an attack scenario can be developed that requires an attack potential of less than 18 per device for identification and initial exploitation or less than 9 for initial exploitation per device as defined in Appendix B, the vendor assertion cannot be verified. At least one attack scenario shall be presented, in a format consistent with the examples shown in Appendix B in this document.

The tester may perform any test needed to validate the attack scenario. The tester shall detail whether steps are based on actual testing or on assumptions and provide justification for any use of assumptions rather than actual testing. Calculations shall include evidence justifying particular rating levels as being appropriate.

The tester shall present evidence of the test methodologies followed and the validation results.

DTR C2.5 Control of any Numeric Interface

The PIN-accepting POI terminal must be equipped with only one payment card PIN-acceptance interface—a keyboard. If another interface is present which can be used as a keyboard, a mechanism must exist to prevent its use for PIN entry—e.g., it must not have numeric keys, or it is not possible to use it otherwise for numeric entry, or it is controlled in a manner consistent with B15.

Guidance

A keyboard may be a touchscreen if the cardholder can input numeric data to it.

Any interface of the device that can be used to accept numeric entry must be controlled.

Devices implementing at the same time a touchscreen for PIN entry and a separate keypad (for example, to meet visual impaired regulations) must meet this requirement.

TC2.5.1 The tester shall visually inspect the device to verify the assertions provided by the vendor in order to verify that:

- The device must be equipped with only one payment card PIN-accepting keyboard, and
- If another interface is present which can be used as a keyboard, a mechanism must exist to prevent its use for PIN entry, for example, it must not have numeric keys, or it must not be possible to use it otherwise for numeric entry or it is controlled in a manner consistent with A8, B15 or C2.4.

TC2.5.2 If the device is equipped with more than one keyboard, and if the keyboard not intended for payment card PIN entry may be operated to accept numeric entry (for example, since it is equipped with numeric keys or since it is fully programmable, like a touchscreen), the tester shall verify that these numeric modes may not be used to enter a payment card PIN, especially that a possible use of numeric entry modes does not interfere with Requirement A8, B15, or C2.4.

DTR Module 3: Communications and Interfaces

DTR D1 Identification of Interfaces

All protocols and all interfaces available on the device are accurately identified by the device vendor. The vendor has a complete and comprehensive understanding of how all protocols and interfaces present on the device interact. All public domain protocols and interfaces available on the device are clearly identified in the* Open Protocols – Protocol Declaration Form*.

Guidance

The objective of this section is to identify all physical interfaces and the corresponding logical protocols that are supported by the device. This is not restricted to only those declared in the Open Protocols – Protocols Declaration Form. It is required that a device vendor know of all interfaces and protocols supported by the device and provide details of these protocols and interfaces within documentation.

Interfaces, which provide for communications, such as but not limited to:

- *Ethernet*
- *Wi-Fi*
- *Audio*
- *Serial*
- *Bluetooth*
- *Cellular (GPRS and CDMA)*
- *USB*

Protocols, such as: but not limited to:

- *PPP*
- *TCP*
- *UDP Light*
- *TLS1.2 or above**
- *HTTP*

The vendor must also identify where they derive their code that is used to implement protocols, as well as document all protocols and which physical interfaces they apply to.

The tester is required to verify the claims of the vendor as described in the asset flow to ensure that the vendor listing of physical and logical interfaces in complete. This requires the tester to both identify physical interfaces through analysis of the design, as well as verify the presence of logical interfaces through methods such as passive monitoring of communications, port/protocol scans, and code review.

Where the interface is supplied by an OEM module:

- *If the module is under the control of the firmware and runs in the same process space as the firmware, the OEM interface module must still be assessed to ensure that secure pairing (for wireless technologies listed above) is provided for and that secure communication is enforced by the interface.*
- *If an independent OEM module is used:*
  - *The protocol and the pairing mechanism must be assessed; and*
  - *The security of the link between the module and the firmware must be assessed.*
- *If the OEM module shares resources with the rest of the device, a vulnerability assessment is required to ensure that the OEM module cannot adversely impact the function of the device.*

(continued)

OEM modules that are found to have unaddressed exploitable vulnerabilities may result in the removal of the entire POI device approval.

Note: *If the device implements an IP stack, the device must support TLS 1.2 or higher.*

TD1.1 The tester shall examine the vendor's asset flow diagrams to verify that they accurately portray all interfaces and protocols present on the device.

TD1.2 The tester shall examine any relevant documentation distributed by the vendor such as schematics, data sheets, asset flow diagrams, and assembly drawings submitted by the vendor to ensure that the vendor has exhaustively defined all protocols and interfaces supported by the device. The vendor must also identify where it derives its code that is used to implement protocols, as well as document all protocols and which physical interfaces they apply to. When public libraries are used to implement protocols, their versions must be specified.

TD1.3 The tester shall complete a table describing all interfaces and protocols; and for each, identify which component implements the protocol, whether it is a security protocol, and the location from which the software was derived. The tester shall Include all exempted protocols and interfaces and note why they have been exempted from OP requirements.

Example Protocol Table

Protocol Name	Component Handling the Protocol	Physical Interface	Source Code Base and Version	Security Protocol	Additional Information
IP (General)	Security Processor		Linux (3.7.1)		
TLS 1.2	Security Processor		OpenSSL (1.0.1g)	XXX	
GPRS	GPRS Modem		Modem vendor		
IP (GPRS)	GPRS Modem		Modem vendor		

TD1.4 The tester shall verify whether the identified protocols and interfaces are in line with the PCI PTS POI Technical FAQs. When forbidden protocol modes or interfaces are included by a standard implementation, the vendor must provide exhaustive documentation describing how these protocol modes or interfaces are disabled by hardware and/ or software means. When changes are performed in the software, the relevant source code must be provided.

TD1.5 The tester shall perform any additional tests necessary to validate the device's documented list of protocols, services, and physical interfaces. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid. The vendor shall provide any necessary test support to access and use the interfaces under test.

DTR D2 Logical Anomalies

The device's functionality shall not be influenced by logical anomalies such as (but not limited to) unexpected command sequences, unknown commands, commands in a wrong device mode and supplying wrong parameters or data, which could result in the device outputting the clear-text PIN or other sensitive data.

Guidance

Functionality shall be considered as any functionality, via any internal or external interface, that could impact the security of all of the device's relevant components.

Vendors should provide software-design rules and specifications to support answers. .

All interfaces and associated communication methods of the device must be assessed to ensure that no interface can be abused or used as an attack vector. This includes the tester enumerating and assessing all logical and physical interfaces regardless of intended usage, without exception. Specifically, this includes any physical, logical, or application interface that is executed within the POI device with sufficient privilege to allow for direct interface to sensitive assets within the POI (should that protocol be compromised in some way). The interfaces must be documented and assessed whether they are used for or have access to card data or not. ***This analysis must be done in accordance with the Appendix G, "Domain-Based Asset Flow Analysis."*** *Sufficient evidence must be provided to demonstrate the validity of laboratory assessments including interfaces using open protocols.*

This includes any Open Protocol communication method that uses a wireless, local, or wide-area network to transport data. This would include but is not limited to Bluetooth, Wi-Fi, cellular (GPRS, CDMA), or ethernet. In addition, any communication that uses a public domain protocol or security protocol would be assessed here.

The vendor shall provide evidentiary matter providing details on internal testing including, but not limited to, the following:

- *Source code reviews targeting specific relevant security-critical functionalities.*
- *Vulnerability analysis; that includes gathering and considering evidence necessary to perform practical testing.*
- *Penetration testing to validate the robustness of the device to protect against feasible attacks by addressing known attack methods. For example (but not restricted to) fuzzing, using appropriate tools and techniques.*
- *Audits of relevant existing test evidence, which may be utilized where appropriate, by giving justifications for validity of evidence and test methodologies overall.*

The laboratory shall determine the veracity of the material provided to determine the degree of reliance that may be placed upon the evidence and, where necessary, the laboratory shall extend the testing.

The device controller is not in scope for this requirement for unattended devices.

TD2.1 The tester shall describe the vendor's measures that ensure the relevant component's functionality is not influenced by logical anomalies such as unexpected command sequences, unknown commands, commands in a wrong device mode, and supplying wrong parameters or data.

TD2.2 The tester shall note the programming languages in which the POI’s firmware source code is written for each of the security processing elements (as detailed in DTR A4).

TD2.3 The tester shall detail the type, version, capabilities, and configuration of the operating system(s) used on each of the POI security-processing elements (as detailed in DTR A4).

TD2.4 Where a complex operating system (such as Linux, other *nix variants, and operating systems such as Android) is used, the tester shall verify that ASLR is enabled and correctly configured. For example, for Linux, this means setting “randomize_va_space” to a value of “2,” and ensuring that all code is correctly compiled (and/or configured) to implement and enable ASLR. Where options are provided for use of stack canaries and data-execution-prevention bits, these must always be enabled (including for application code).

TD2.5 The tester shall enumerate all logical and physical interfaces provided by the POI. For example, the tester shall include physical interfaces such as USB input, serial input, IC interface, TCP/IP interface, etc., as well as API interfaces provided to applications that may execute on the POI.

TD2.6 If the device includes command-execution interfaces or parsers: The tester shall detail how each of the interfaces identified in D1 is configured to accept commands—for example, whether a command executive is used, or other methods are used to parse input commands. The tester shall define which common functionalities exist between interfaces to determine which test approaches may be applied in common to more than one interface.

TD2.7 The tester shall detail in an appendix to the evaluation report a complete list of all APIs as defined by the vendor that are provided on each of the logical interfaces of the POI.

TD2.8 The tester shall perform a source-code review of each relevant interface and confirm that it is handled securely, that only documented commands are implemented, and that secure defaults are provided for each interface. The tester shall detail the methods used to verify the length and content of each command before processing. The tester shall derive and describe vulnerability-analysis models from source-code review and other available evidence to determine attack paths and appropriate penetration testing.

The source code review should be targeted on relevant security-critical functionalities such as (but not restricted to): buffer overflows; unhandled exceptions, read-access violations, and denial-of-service conditions, etc., including factors that are specific to the POI’s OS, communications protocols, and source code software language(s).

TD2.9 For systems that are designed to execute non-firmware applications, the vendor shall provide a test application to be run into the device containing a buffer-overflow vulnerability, together with the application’s source code. The tester shall run the application and determine if the device fails securely.

TD2.10 The tester shall identify and review publicly available sources of vulnerability disclosure, including but not necessarily limited to those referenced in the vendor-certification process document and shall check that any vulnerabilities found cannot be exploited on the POI. The tester shall execute a vulnerability assessment and/or testing, to identify vulnerabilities associated with the interfaces and associated communication methods of the device.

The tester shall compare his/her analysis with the analysis provided by the vendor and confirm that the vendor has remediated any problems with these vulnerabilities, and that firmware-audit report documents exist and have been updated to validate this assertion.

TD2.11 The tester shall perform fuzzing of the POI security-relevant interfaces in order to uncover logical anomalies.

The tester shall describe fuzzing methodologies and tools used, as well as any assistance provided by the vendor in the case of white-box fuzzing and results obtained. The description and results should be sufficient to provide a demonstrably high level of confidence in the security of the POI interfaces. This shall include examples of work performed with explanations for selecting certain test strategies and fuzzing boundary conditions.

The methodology used should provide an optimum balance between the use of evaluation resources for penetration testing and reliance upon the vendor-provided information.

The tester shall provide any evidence and descriptions necessary to support conclusions.

The evaluation may rely upon appropriate testing results provided by the vendor, which shall be less than two years older than the date of the current evaluation submission, in order to replace/supplement the laboratory testing.

The vendor's testing findings and testing methodology should be documented.

The tester shall describe the methodology and the findings of the vendor's testing, which must be sufficient to demonstrate that are at least as robust as the criteria described here.

TD2.12 The tester shall identify all command interpreters within the firmware—e.g., anything scriptable, including but not limited to SQL commands and OS commands (for example, UNIX system() and popen()). If command interpreters are called, the tester shall describe and justify why a command or environment cannot be manipulated to perform unauthorized functions.

TD2.13 The tester may perform any additional tests necessary to validate the device's compliance to this DTR. The vendor shall provide any necessary test support to access and use the interfaces under test.

DTR D3 Security Guidance for the Protocols and Interfaces

The device has security guidance that describes how protocols and interfaces must be used for each interface that is accessible by the device applications.

Guidance

The objective of this section is to verify the documentation details how each interface and protocol is configured and integrated with the application. It should be clear which interfaces and protocols can be used for applications. The guidance must make sure that application developers are guided in the secure use of protocol or interfaces.

The vendor must also identify and document all protocols and Interfaces identified in Section D1.

TD3.1 The tester shall examine and assess the distribution process of the security guidance. It must ensure that all necessary users are aware of it and have access to the latest version. The vendor must specify what is the accessibility scope of each document—i.e., indicate which documents are intended as public and which documents have a restricted audience.

TD3.2 The tester shall summarize all the protocols and interfaces provided by the firmware to application developers. The tester shall ensure that each of these protocols and interfaces have associated guidance within vendor documentation that ensures that application developers are guided towards secure use of any interface or protocol.

DTR D4 Default Configuration of the Interfaces

The device has guidance that describes the default configuration for each protocol relevant to each interface that is available on the device. Each interface and protocol on the device should be configured with secure default settings.

Guidance

The objective of this section is to verify the default configuration for interfaces and communications as defined by the vendor documentation and Section D1.

The lab must assess that the default configuration for each interface is not in an unsecure state and that non-essential services are disabled.

TD4.1 The tester shall examine any relevant documentation—such as a user guide, the application developer's guide, design specifications, the interface specification, the software-design rules and specifications, or the implementation guidance—and confirm that each interface defaults to a secure or disabled state.

TD4.2 The tester shall assess the default configuration of the device. The default configuration must be in line with security guidance—for example, default settings must disable non-essential services, must use secure configurations for security protocols.

TD4.3 The tester shall perform any additional tests necessary to validate the device's documentation. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid. The vendor shall provide any necessary test support to access and use the interfaces under test.

DTR D5 Key Management Security Guidance

The device has guidance for key management describing how keys and certificates must be used.

a) ***The key-management guidance is at the disposal of internal users and/or of application developers, system integrators, and end-users of the device.***

b) ***Key-management security guidance describes the properties of all keys and certificates that can be used by the device.***

c) ***Key-management security guidance describes the responsibilities of the device vendor, application developers, system integrators, and end-users of the device.***

d) ***Key-management security guidance ensures secure use of keys and certificates, including certificate status—e.g., revoked—secure download, and roll-over of keys.***

Guidance

Note: *This does not supersede any criteria in B9 but rather is required for any device implementing protocols evaluated under Open Protocols—i.e., key-related Security Protocols, such as SSL/TLS, SSH, VPN technologies.*

This requirement applies to all declared Security Protocols defined in Section D1.

TD5.1 The tester shall examine the key-management security guidance, which must:

- Cover all the keys and certificates used by the security protocols;
- Cover all key properties, minimally type and length—e.g., TDES 112-bits;
- Cover all users of keys and certificates. If different users are involved, responsibilities must be clearly indicated;
- Be complete, clear, and unambiguous; and
- Ensure that keys are not shared between security protocols.

Note: *If, because of device life cycle and architecture, the vendor does not perform key management, the vendor must provide sufficient guidance for the device user to create effective key-management policies and procedures.*

TD5.2 The tester shall verify whether the cipher suites used by the declared security protocol are in line with the PCI PTS POI Technical FAQs. The tester shall perform any additional tests necessary to validate the device's documentation.

TD5.3 The tester shall provide a key table for Open Protocol-related keys and passwords or passphrases used to derive keys. This table must include all OP keys and passwords stored in persistent storage. Session keys used temporarily may optionally be listed in the table.

Example Key Table

Key Name	Purpose/Usage	Algorithm	Size (Bits)	Managed by	Storage
TLS Root Keys (public keys)	Authentication of certificate chain from TLS server	RSA, DSA or ECDSA	…	Application	Application
TLS Client Keys (private keys)	Providing cryptographic authenticity of client to the server	RSA, DSA or ECDSA	…	Application	Application
Bluetooth Long-Term Key (LTK)	Establishing a secure channel with a Bluetooth peer	E0 or AES	128	Firmware	Flash File System
Wi-Fi Passphrase	Derivation of encryption key for Wi-Fi traffic	PBKDF used to create AES 256 key	..	Application	Application

DTR D6 Use of Secure Protocols

The device has all the security protocols that are available on the device clearly identified in the* Open Protocols – Protocol Declaration Form. *The device vendor provides documentation that describes the implementation and use of the security protocols that are available on the device.

Guidance

The declared security protocol is able to ensure confidentiality, integrity, server authentication, and protection against replay.

For D7, D8, and D9, the minimum requirements for cryptographic algorithms used to provide security are specified in DTR B9. Only TDES, RSA, ECC, DSA, and AES are acceptable for encryption or signing operations. SHA256 or above may also be used for hashing purposes. See Appendix E.

TD6.1 The tester shall review vendor documentation and describe how the documentation clearly defines the declared security protocol for each interface.

TD6.2 The tester shall assess the physical and logical interfaces of the device to ensure the declared security protocol is present for each interface.

TD6.3 The tester shall execute tests to verify that each interface can use a declared security protocol in a secure way. The tester may perform any additional tests necessary to validate the device's documentation. The tester should use his or her own judgment in determining appropriate tests. The vendor shall provide any necessary test support to access and use the interfaces under test. The tester shall summarize the testing and responses for each interface tested.

DTR D7 Secure Protocols to Provide Data Confidentiality

As defined in the asset flow diagrams, the device is able to provide confidentiality of data sent or received over a network connection.

a) ***Encryption mechanism utilizes key sizes appropriate for the algorithm(s) in question.***

b) ***Encryption is provided by using keys that are established in a secure manner using appropriate key-management procedures, such as those listed in*** **NIST SP800-21, Guideline for Implementing Cryptography in the Federal Government,** ***and*** **ISO 11568 Banking – Key Management (Retail).**

Guidance

The intent of this requirement is to ensure that any claims of a security protocol are effective in meeting PCIs cryptographic requirements for sufficient confidentiality.

Refer to PCI guidance for appropriate cryptographic algorithms and key sizes.

TD7.1 The tester shall review vendor documentation and describe how the declared security protocol for each interface provides data confidentiality.

TD7.2 The tester shall execute tests necessary to validate the device's implementation of a security protocol is providing for confidentiality of data in line with the documented method. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid. The vendor shall provide any necessary test support—including testing keys and/or certificates and tools to load them on the device—to access and use the interfaces under test. The tester shall summarize the testing and responses for each interface tested.

DTR D8 Secure Protocols to Provide Data Integrity

As defined in the asset flow diagrams, the device is able to provide the integrity of data that is sent or received over a network connection.

a) ***Integrity is provided by a MAC as defined in* ISO 16609*, or by a digital signature.***

b) ***Hashing can be provided by at least one of the following algorithms: SHA-224, SHA-256, SHA-384, and SHA-512.***

c) ***Examples of appropriate algorithms and minimum key sizes are stated in Appendix E of the PCI PTS POI DTRs.***

Guidance

The intent of this requirement is to ensure that any claims of a security protocol are effective in meeting PCIs cryptographic requirements for sufficient data integrity.

Refer to PCI guidance for appropriate cryptographic algorithms and key sizes.

TD8.1 The tester shall review vendor documentation and describe how the declared security protocol for each interface provides data integrity.

TD8.2 The tester shall verify device behavior when receiving incorrect data packets.

TD8.3 The tester shall perform any additional tests necessary to validate the device’s implementation of security protocol is sufficiently able to provide integrity. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid. The vendor shall provide any necessary test support to access and use the interfaces under test. The tester shall summarize the testing and responses for each interface tested.

DTR D9 Secure Protocols to Provide Mutual Authentication

As defined in the asset flow diagrams, the device uses a declared security protocol and support mutual authentication.

a) Server authentication utilizes key sizes appropriate for the algorithm(s) in question, as denoted in Appendix E.

b) Hashing can be provided by at least one of the following algorithms: SHA-224, SHA-256, SHA-384, and SHA-512.

c) The device is able to verify the validity of the public keys it receives.

d) The device is able to verify the authenticity of the public keys it receives.

e) The device's trusted root certificate store shall contain only public key certificates from trusted CA's or else self-signed certificates verified by the acquirer.

TD9.1 The tester shall review vendor documentation and describe how the declared security protocol provides mutual authentication.

TD9.2 In the case when certificates are used for server authentication, **t**he tester shall execute tests to verify device behavior when receiving incorrect certificates, including:

a) Expired certificates

b) Self-signed (un-authenticatable) certificate

c) Certificate with weak key size—e.g., RSA less than 2048 bits

d) Certificate signed using a weak hash—e.g., SHA1 or MD5

e) Chaining error in certificate for cases a, b, c, or d

TD9.3 The tester shall verify the device's behavior when connecting to an un-authenticated server. The tester shall verify that the connection is rejected. If the default behavior is to accept the connection without device authentication, documentation must exist to strongly suggest mutual authentication is enabled. One example could be achieving mutual authentication in a higher-level protocol, such as an application level one.

TD9.4 The tester shall perform any additional tests necessary to validate the device's implementation of the security protocol is able to provide authentication. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid. The vendor shall provide any necessary test support to access and use the interfaces under test.

DTR D10 Secure Protocols to Provide Exception Handling and Replay Detection

As defined in the asset flow diagrams, the device is able to detect replay of messages and enables the secure handling of the exceptions.

TD10.1 The tester shall review vendor documentation and describe how the declared security protocol for each interface provides exception handling and replay detection.

TD10.2 Unless it can be shown that by design the used security protocol provides message replay countermeasures (by the usage of counters or nonces, for example) and that there is no publicly known replay attack on the protocol or its implementation under evaluation, the tester shall test to verify device behavior when receiving incorrect data packets, replay messages, or other anomalies.

TD10.3 The tester shall perform any additional tests necessary to validate the device's implementation of security protocol is able to provide exception handling and replay detection. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid. The vendor shall provide any necessary test support to access and use the interfaces under test. The tester shall summarize the testing and responses for each interface tested.

DTR D11 Session Management

As defined in the asset flow diagrams, the device implements session management.

a) ***The device keeps track of all connections and restricts the number of sessions that can remain active on the device to the minimum necessary number.***

b) ***The device sets time limits for sessions and ensures that sessions are not left open for longer than necessary.***

TD11.1 The tester shall review vendor documentation, confirm whether the declared security protocol for each interface provides session management, and describe how this is provided.

TD11.2 The tester shall assess each interface of the device to ensure that each interface can provide session management.

TD11.3 The tester shall verify device behavior while attempting to break session management rules—e.g., request to exceed number of simultaneous connections.

TD11.4 The tester shall perform any additional tests necessary to validate the device's implementation of security protocol is able to provide session management and that the device handles any anomalies correctly. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid. The vendor shall provide any necessary test support to access and use the interfaces under test. The tester shall summarize the testing and responses for each interface tested.

DTR D12 Bluetooth Security

Bluetooth communications must be secured against eavesdropping and man-in-the-middle attacks.

Note: If Bluetooth is used, D14 may alternatively be used.

Guidance

If a Bluetooth interface is used, the Bluetooth interface must enforce encryption. This encryption is in addition to any other encryption the data may have undergone. If PIN or passkey entry is to be used, the evaluator must validate that vendor default values can be changed. The device must not support or allow for the use of insecure communication options such as, but not limited to, Security Modes 1 and 2 and the "Just Works" secure pairing option of Security Mode 4.

For Bluetooth 4.1 or higher, devices that have BR, EDR, and High Speed (HS) features, Security Mode 4, Level 4 must be used. This requires Secure Connections, which uses authenticated pairing and encryption using 128-bit strength keys generated using FIPS-approved Advanced Encryption Standard (AES) encryption. For Bluetooth 2.1 through 4.0 devices, Security Mode 4, Level 3 must be used.

BLE implementations must use version 4.2 or higher. BLE must use LE Security Mode 1 Level 4 (Secure Connections) only—Just Works cannot be used at any time. The device must not support or allow for the use of insecure communication options such as, but not limited to, LE Security Mode 2, and levels 1, 2 and 3 of LE Security Mode 1 and the "Just Works" secure pairing option of Security Mode 1. This must be documented in the security policy made available on the PCI website.

BLE implementations in SCRPs, regardless of Bluetooth version, for use in SPoC/MPoC Solutions may use unauthenticated pairing (Just Works) provided compensating controls to mitigate against eavesdropping and MITM attacks are in place as part of the Solution. These controls shall be validated during testing of the SPoC/MPoC solution. SCRPs that allow either the use of Just Works for pairing or do not exclusively implement Secure Connections are not approved for use outside of a SPoC/MPoC Solution.

TD12.1 The tester shall describe the Bluetooth versions and pairing modes that are supported by the device.

TD12.2 The tester shall describe the physical configuration used for Bluetooth. This will include any Bluetooth adapters, communication links, and Bluetooth software executing on the main processor/s.

TD12.3 The tester shall set up the device and confirm that the connections as specified above are operable.

TD12.4 The tester shall try to attempt to connect to the device using non-supported modes, including no encryption, Just Works, or BLE 4.0 Legacy Pairing.

TD12.5 The tester shall describe the operator interaction during pairing and how authentication is enforced—for example, does the device show a number on its display, accept a number via a keypad, or use out-of-band authentication etc.

DTR D13 Wi-Fi Security

Wi-Fi communications must be securely configured. Protocols with known vulnerabilities must be disabled.

Note: If Wi-Fi is used, D14 may alternatively be used.

Guidance

If a Wi-Fi interface is used, the Wi-Fi interface must enforce encryption. This encryption is in addition to any other encryption the data may have undergone. Security must be enabled. WEP cannot be used or configured at any time, and WPA2 and/or WPA3 must be supported. If passkey is used, it must not be a vendor default. The evaluator must validate that default values can be changed on the target of evaluation.

TD13.1 The tester shall describe the different protocols and modes supported by the device

TD13.2 Using a test system, the tester will confirm that the device does not support WEP.

TD13.3 The tester shall describe how the passkey is entered and whether any default values are handled.

DTR D14 Interface Isolation

Wireless communication interfaces which do not have specific security requirements, or have not met those requirements as listed, must be physically or cryptographically isolated.

Note: Where the applicable security requirements D12 and/or D13 for Bluetooth or Wi-Fi are not met, D14 must be used.

Guidance

This requirement is intended to cover wireless interfaces that are primarily designed for communications. Interfaces such as displays, cameras, speakers and microphones, and other types of transmission and receiving systems that are not primarily designed for data communication between two electronic systems are not in scope for this requirement. Wireless interfaces implemented primarily for payment acceptance, such as NFC or contactless interfaces, are also excluded.

Wireless interfaces are of particular concern because they allow for remote attack vectors or potential public access to the communications. This can allow for physically distant monitoring of communications or deployment of logical attacks onto one or more terminals from a remote location. To mitigate these risks, this standard has specific security requirements for some common wireless interfaces such as Bluetooth and Wi-Fi. Where a wireless interface exists with no such specific requirements, or where requirements do exist but are not met by the POI implementation, that interface must be considered an insecure interface. Cellular communications, where the network provider uses a SIM or similar mechanism to authenticate the wireless connection, are excluded from the scope of this requirement.

Interfaces in scope for this requirement must be protected using cryptography and may be required to be additionally isolated using hardware separation, depending on how the interfaces are used and what data is processed. Systems that manage only PAN data or are only able to obtain card data from a payment instrument that provides its own authentication data (such as an EMV payment card) may implement cryptographic protection only for an insecure interface.

Insecure interfaces are assumed to be open to compromise; therefore, measures must be implemented to ensure they are not able to expose clear-text sensitive data and that the execution environments in which that data is processed are protected. It is not sufficient to ensure that data transmitted across such an interface is simply encrypted if there are not also authentication controls to prevent injection of malicious data.

Therefore, cryptographic protection must provide methods to protect both the authenticity and confidentiality of the data transmitted and received on that interface. Connections using recent versions of TLS—i.e., TLS 1.2 or higher—with cipher suites that implement strong cryptography are an example of an acceptable cryptographic protection method. Alternatively, SRED controls may be used if these meet the requirements to be always enabled and provide authentication across transmitted data.

Cryptographic protections must be enforced and set to "always on" by the firmware of the device. It is not compliant to have a system that provides functions to provide cryptographic controls but leaves the configuration and use of those controls to the applications executed within the POI.

(continued)

Systems that accept card data and/or customer PINs must implement hardware isolation for insecure interfaces, in addition to cryptographic isolation. These systems must ensure that the interface code and processing of communications data are implemented on a physically separate execution environment to that of the card data and/or customer PINs processing. An example of an acceptable implementation would be to have card data and/or customer PINs processing implemented on a physically separate processor to the interface, with clearly defined APIs between the two processors.

Hardware isolated systems must ensure that sensitive data is encrypted prior to being sent to the interface processor, although controls for the authenticity of the connections may be applied at the interface processor. For example, it would not be compliant for a system to pass unencrypted payment card data to the interface processor, where it was then encrypted using TLS for communication across the insecure interface. However, it may be acceptable for the data to be passed to the interface processor encrypted—using SRED, for example—and then the interface processor implements a TLS connection to validate the authenticity of the transmission endpoint.

Additionally, application of SRED controls that provide both confidentiality and authenticity controls, or the use of both SRED and TLS, prior to transmission to the interface processor may also be compliant.

TD14.1 The tester shall detail all wireless communication interfaces supported by the device, such as Wi-Fi, Bluetooth, cellular, bespoke wireless, etc. Wireless interfaces not primarily designed for communication between two electronic systems, such as cameras, microphones, and displays, are not in scope of this requirement. Contactless and NFC interfaces are also considered out of scope of this requirement.

TD14.2 The tester shall note which interfaces have specific security requirements within this program (within this standard, or detailed externally in FAQ documents), and whether all of those interfaces meet the requirements as defined. Interfaces not excluded through these first two tests must be validated against the remaining requirements.

TD14.3 The tester shall confirm that the POI implements cryptographic controls over the data communicated across any interface in scope of this requirement. The tester shall detail how these cryptographic controls are used, what algorithms and key sizes may be configured, and how it is ensured that the controls are always active on the wireless interfaces.

TD14.4 The tester shall confirm that the POI does not implement any open ports or listening services—such as FTP, SSH, or HTTPS—running over wireless interfaces in scope of this requirement. All connections must be initiated by the POI and protocols are assessed under the applicable protocols or interface requirements. The POI must not be able to be used or implemented as an access point that provides wireless connections, other than in a peer-to-peer system like Bluetooth. In such systems, connections must only be permitted for transaction-processing purposes, such as to an external communications device or printer.

TD14.5 For systems relying on SRED for cryptographic protections, the tester shall confirm the system has been assessed to be compliant to the SRED requirements. Additionally:

a) The testers shall confirm that the SRED encryption features are always active and cannot be disabled temporarily or permanently, or the device deployed in a way that SRED may not be activated.

b) For systems relying solely on SRED features, the tester must also confirm that authenticity is applied to all connections implemented across the insecure interface through the use of a (H)MAC, CMAC, or digital signature.

TD14.6 The tester shall confirm that the cryptographic controls, including SRED controls if they are relied upon for compliance, cannot be disabled or configured into a non-compliant state by the application. Any cryptographic controls relied upon for compliance to this requirement must be managed and controlled by the device firmware and the POI must be deployed with these in an “always on” state. It is not acceptable to rely upon guidance requiring an application developer to ensure all account data is encrypted before being output.

TD14.7 The tester shall confirm how the cryptographic controls ensure that all data transmitted across the wireless interface(s) is protected against eavesdropping.

TD14.8 The tester shall confirm how the cryptographic controls ensure that all connections across the wireless interfaces are cryptographically authenticated. This authentication must cover all data transmissions and allow for the authentication of the receiving end point in any connection that negotiates keys as part of the initiation of the connection (such as TLS).

TD14.9 The tester shall confirm if the POI allows for the input or processing of customer PINs or card data. Where this is the case, the tester shall detail how the wireless interfaces are physically isolated from the card data and PIN processing elements. Use of a separate processor for implementing the interface is an example of acceptable physical isolation

TD14.10 The tester shall confirm that systems that allow for the input or processing of customer PINs or card data ensure that the assessed cryptographic protections are applied prior to transmission to the physically isolated processing area that implements the wireless interface.

DTR Module 4: Life Cycle Security Requirements

E – During Manufacturing

Compliance can be proven by evidence that the procedures comply by design via:

- Evidence of existence—i.e., that these procedures are implemented. Example: The lab has received procedures from the company that implements the requirement.
- When these procedures are in use, samples of the process in operation. For this purpose—in a timeframe of 3, 6, or 12 months—samples can be randomly collected and validated. Example: For E3, the lab can randomly select a few occasions and ask for the related log belonging to the dual control or standardized cryptographic authentication procedure used.

DTR E1 Change Control

Change-control procedures are in place so that any intended change to the physical or functional capabilities of the POI causes a re-certification of the device under the impacted security requirements of this document. Re-certification is not required for changes that purely rectify errors and faults in software in order to make it function as intended and do not otherwise remove, modify, or add functionality that impacts security. Approval of delta submissions is contingent on evidence of the ongoing change-control and vulnerability management process.

Guidance

The organization must utilize a documented secure change management system, such that all development is traceable and access restricted. All changes should be reviewed and approved by cognizant management prior to release. The change-control procedures shall include the unique identification of all configuration items and their developer, including the device, its parts (such as ICCR, MSR, and keyboard) and its firmware.

The vendor must have a mechanism that allows for the identification of the exact parts (bill of materials or source code files) for a deployed device. For hardware, this mechanism may use the printed hardware version number or another field such as serial number or production date. For firmware, the vendor must be able to use a displayed build ID or equivalent to determine the exact source code, libraries, and firmware-related build settings.

Note: *Hardware and firmware version number identifiers may consist of a combination of fixed and variable alphanumeric characters, whereby a lowercase "x" is used by PCI to designate all variable fields. The "x" represents fields that the vendor can change at any time to denote a different device configuration. Options that cannot be a variable character include those that directly pertain to meeting security requirements.*

The hardware version number as printed on the label need not represent the complete bill of materials for the manufactured device. Where changes to the bill of materials affect security, then this necessitates a change to the non-wildcarded fields on the printed hardware version number.

Pure bug fixes, including patches for known security flaws, do not require a delta submission or a change to the non-wildcarded firmware version number fields.

TE1.1 The tester shall examine and cite any supporting documentation submitted by the vendor and verify the documentation details change-control processes and procedures for physical and functional changes to the POI device, describing how the items are uniquely identified such that it is possible to track the different versions of the items, and including criteria for when changes are submitted for PCI evaluation and details of the vulnerability management process.

TE1.2 The tester shall describe how the vendor can recover a list of all components—i.e., exact components in the bill of materials or exact source code—used in the hardware and firmware of a particular device.

TE1.3 The tester shall confirm that the POI devices are labeled with the unique reference assigned within the change-control procedures. The tester shall also confirm that the tamper-resistant parts—e.g., PED, ICCR—have a unique identifier that is visible without opening the devices. The tester shall confirm that the firmware is uniquely identified.

TE1.4 The tester shall examine a sample of documentation of physical and functional changes to POI devices, such as change-control logs, to validate that procedures are followed, including the submission and sign-off processes.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TE1.5 The tester shall interview those responsible for the change-control processes—including engineers and programming staff, supervisory staff, technical management, etc.—to verify that the documented and approved procedures are known and understood by all affected parties.

TE1.6 The tester shall validate that either:

- All changes to PCI approved devices have been submitted for re-evaluation; or
- Only changes that purely rectify errors and faults in software in order to make it function as intended and do not otherwise remove, modify, or add functionality are deferred from submission and not submitted immediately, but are instead submitted on an aggregate basis.

DTR E2 Firmware Certification

The firmware and any changes thereafter have been inspected and reviewed using a documented and auditable process and verified by the vendor as being free from hidden and unauthorized or undocumented functions.

Guidance

Firmware is considered to be any code within the device that provides security protections needed to comply with PCI requirements. This is true regardless of how labelled—e.g., OS, EPROM code, DLL's, parameter files, applications, kernel code. This includes any code that is not in the Vendor Domain as defined in Appendix G, "Domain-Based Asset Flow Analysis." Other code that exists within the device that does not provide security, and cannot impact security, is not considered firmware under PCI requirements, except for SCRs intended for use with COTS devices and SCRPs, where all code is considered firmware.

"Certify firmware" refers to self-certification. This requirement, in essence, requires the vendor to have implemented and to use internal quality control and change-control systems. With these systems in place, the vendor is in control of the code and can attest to the fact that the code is free of hidden or unauthorized functions.

The vendor shall indicate the compiler settings used in order to maximize the mitigation of known vulnerabilities.

The vendor shall implement measures to help prevent common exploits of "buffer overflow" and similar vulnerabilities. Strategies by the developer to address these vulnerabilities may include:

- *Avoiding them by design (extreme example: using a programming language, which prevents buffer overflow by definition);*
- *Finding them by adequate testing (for example, static code analysis or comprehensive fuzz testing); and/or*
- *Mitigating them by techniques that include but are not limited to: Address Space Layout Randomization (ASLR), Data Execution Prevention (DEP), Harvard Architecture, and Stack Canaries.*

The vendor shall document where labs may place reliance upon this in connection with D2 and other relevant requirements.

TE2.1 The tester shall examine details provided by the vendor that the documented process explicitly addresses how testing/auditing has been carried out to check for unauthorized and undocumented functions.

TE2.2 The tester shall detail any compiler settings used by the vendor in order to maximize the mitigation of known vulnerabilities. If no specific compiler settings are used, the tester shall detail how known vulnerabilities are mitigated.

TE2.3 The tester shall detail any mitigation techniques used by the vendor to help prevent common exploits. The tester must state and justify any reliance placed on these technique(s) in minimizing testing. If no specific techniques are used, the tester shall detail how the common exploits are prevented.

TE2.4 The tester shall confirm that—and summarize how—the POI vendor has a documented software-development process that details how firmware must be written, reviewed, and tested to ensure the firmware is free from security vulnerabilities.

The tester shall confirm that—and summarize how—the POI vendor has a documented software-development process that details how firmware developers need to be trained to recognize and avoid common security problems.

The tester shall reference the names and versions of all relevant documents that define the software-development process.

TE2.5 The tester shall confirm that—and summarize how—the documented software-development process provides specific guidance for programmers, reviewers and testers, and does not rely on non-specific statements (for example, the guidance "does not create buffer overflows" would be insufficient as it does not provide information to the programmer on how to prevent these problems.

TE2.6 The tester shall confirm that—and summarize how—the certification process includes checking of all code that executes in all processing elements as listed in DTR A4.

TE2.7 The tester shall confirm that—and summarize how—the process described above includes checking sources of vulnerability disclosure (such as the national vulnerability database) for public vulnerabilities that may apply to the POI firmware.

TE2.8 The tester shall confirm and describe, via documentation review, that the certification process requires that the code review and security testing is performed by a person who was not involved in the authorship of the POI code.

TE2.9 The tester shall confirm and describe, via documentation review, that the certification process requires that the code review and security testing be performed after every security-relevant change, and before the firmware is released to production.

TE2.10 The tester shall confirm and describe, via documentation review, that the certification process is designed to result in an auditable process that shows the code review and security testing have been performed and requiring sign-off by the person(s) performing the code review and security testing. The tester shall confirm that the process will show any problems noted during the code review and security testing.

TE2.11 The tester shall obtain, review, and summarize the firmware-verification audit results provided by the vendor for the version of firmware submitted for evaluation. The tester shall confirm that this shows that no problems were found during review of this firmware (or that any problems found have been addressed).

TE2.12 The tester shall review previous firmware-verification audit reports provided by the vendor and summarize these reports and their findings. The tester shall ensure that the reports sampled include security problems found during the review and confirm that these problems have been addressed. If all audit reports reviewed indicate that no security problems have been found, the tester shall justify why it is possible that the firmware image is, and has always been, completely free of security defects.

TE2.13 The tester shall detail from the vendor-provided documentation the process used to manage firmware from development, through certification, to release to production, and installation into the POI. The tester shall justify how this process ensures that it is infeasible for the code image to be altered after it has been reviewed and signed off during the firmware-verification process.

TE2.14 The tester shall outline the following information: If the firmware is based on a general-purpose operating system (like Linux or Windows CE), the steps described in TE2.13 hold for this operating system. The documentation provided by the developer shall show that state-of-the-art rules for "hardening" the operating system have been applied. For example, the developer should provide a table showing a list of all known issues (like patch levels; not including unused packages, etc.; not being able to log into the operating system without cryptographic authentication in operational mode of the POI; not being able to use debug functions like "netdump" during operational use) with remarks indicating how each issue has been addressed for the firmware under evaluation. Similar steps need to be done for other firmware parts that are taken from external sources. The evidence provided by the developer should also include acceptance procedures, showing how the developer ensures that external software can be considered secure.

DTR E3 Certified Firmware Control

The certified firmware is protected and stored in such a manner as to preclude unauthorized modification during its entire manufacturing life cycle—e.g., by using dual control or standardized cryptographic authentication procedures.

Guidance

All certified firmware must be signed prior to distribution and signed using dual control. It should be managed such that no single person is able to sign files. If two secrets (passwords/authentication codes) are required for operation of the signing tool, no single person should know both secrets. All certified firmware must be reviewed against the organization's coding standards prior to signature.

TE3.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail firmware control and validation processes and procedures for storage during the manufacturing process.

TE3.2 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail dual control and cryptographic authentication processes and procedures during manufacturing and describe how this shows compliance to this requirement.

TE3.3 The tester shall examine a sample of documentation for firmware authentication—i.e., signing—and storage during manufacturing, including change-control logs and firmware validation procedures to ensure procedures are followed.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TE3.4 The tester shall interview those responsible for the firmware control processes—including engineers and programming staff, peer reviewers, supervisory staff, technical management, etc.—to verify that the documented and approved procedures are known and understood by all affected parties.

TE3.5 If firmware signing is done on site, the tester shall observe the signing process, the signing tools, and ensure they are under dual control and that the signing procedures are followed.

TE3.6 The tester shall observe the firmware storage and validation process to ensure that only signed and valid firmware is used during manufacturing.

DTR E4 Device Hardware Component Control

The device is assembled in a manner that the hardware components used in the manufacturing process are those hardware components that were certified by the PIN Entry and/or POI Terminal Integration Security Requirements evaluation, and that unauthorized substitutions have not been made.

TE4.1 The tester shall examine and cite any supporting documentation submitted by the vendor and describe how this shows compliance to this requirement. The documentation should detail the hardware components used and the hardware-component control processes and procedures for storage and verification during the manufacturing process including hardware-component identification verification, with the procedures checked in TE1.

TE4.2 The tester shall examine and cite any supporting documentation submitted by the vendor and describe how this shows compliance to this requirement. This shall include:

- Describing how it is ensured that the hardware components used in assembly are the same as those used in the devices submitted for certification.
- Providing details on how hardware components are stored before being assembled into devices.
- Describing how the manufacturer verifies parts before using them in manufacture.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TE4.3 The tester shall interview those responsible for hardware-component control processes—including engineers and line staff, supervisory staff, technical management, etc.—to verify that the documented and approved procedures are known and understood by all affected parties.

TE4.4 The tester shall examine the device parts listing and sample hardware components during manufacturing, to ensure the correct components are used.

TE4.5 The tester shall observe hardware-component control to ensure that authorized components are verified prior to manufacturing and that unauthorized substitutions are not made.

DTR E5 Production Firmware Control

Production software—e.g., firmware—that is loaded to devices at the time of manufacture is transported, stored, and used under the principle of dual control, preventing unauthorized modifications and/or substitutions.

Guidance

> *All software (e.g., firmware) loaded into the device must be signed prior to use and all software (e.g., firmware) needs to be controlled and protected during manufacturing. All software (e.g., firmware) must be validated before each use to prevent unauthorized modifications and/or substitutions.*

TE5.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail firmware control processes and procedures for loading firmware during the manufacturing process.

TE5.2 The tester shall examine, cite, and reference any supporting documentation submitted by the vendor. The documentation should detail firmware control processes and procedures for transporting and storing firmware during the manufacturing process and that the principle of dual control is followed to prevent unauthorized modifications and/or substitutions.

TE5.3 The tester shall examine a sample of documentation for firmware control and storage during manufacturing, including change-control logs and firmware validation procedure to ensure the principle of dual control is followed to prevent unauthorized modifications and/or substitutions.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TE5.4 The tester shall interview those responsible for the software (e.g., firmware) control processes—including engineers, software developers, line staff, supervisory staff, technical management, etc.—to verify that the documented procedures are known and understood by all affected parties.

TE5.5 The tester shall observe the firmware storage and validation process to ensure that procedures are followed during manufacturing.

DTR E6 Post-Production Storage

Subsequent to production but prior to shipment from the manufacturer's or reseller's facility, the device and any of its components are stored in a protected, access-controlled area or sealed within tamper-evident packaging to prevent undetected unauthorized access to the device or its components.

Guidance

Completed devices and any of their components must be controlled and stored in tamper-evident packaging and/or stored in an accessed controlled area until shipped. The device or components must be checked prior to shipping to ensure the device has not been modified or tampered.

This requirement applies to all devices, or parts thereof, that are vulnerable to tampering after manufacturing but before delivery to the customer. Manufactured devices that have tamper detection and response enabled are exempt from this requirement.

TE6.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail documented and approved post-production control processes and procedures for storage and validation of devices or their components subsequent to production but prior to shipping.

TE6.2 The tester shall examine, cite, and reference any supporting documentation submitted by the vendor. The documentation should detail tamper-evident packaging or the access-controlled area where devices and components are stored prior to shipping.

TE6.3 The tester shall examine a sample of documentation for post-production storage, including inventory logs, shipping documentation, and device-validation procedures to ensure procedures are followed.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TE6.4 The tester shall interview those responsible for the post-production control processes—including engineers, software developers, line staff, supervisory staff, technical management, etc.—to verify that the approved and documented procedures are known and understood by all affected parties.

TE6.5 The tester shall observe the device and component storage and validation process to ensure that procedures are followed subsequent to production.

TE6.6 The tester shall examine the vendor's tamper-evident packing or access-controlled area to ensure unauthorized access to devices or their components is not possible.

DTR E7 Secret Information

The device must be authenticated at the key-loading facility or the facility of initial deployment by means of secret information placed in the device during manufacturing. This secret information is unique to each device, unknown and unpredictable to any person, and installed in the device. Secret information is installed under dual control to ensure that it is not disclosed during installation, or the device may use an authenticated public-key method.

Authentication by secret information is mandatory in POI v6.

Guidance

One example of such information is the private key of an asymmetric key pair with the public key of the device signed by a private key known only to the manufacturer.

Dual control is not required if the device generates the secret information and never outputs it, for example if it generates a public/private key pair and never outputs the private key.

TE7.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail that the device will be authenticated at the key-loading facility or the facility of initial deployment by means of secret information placed in the device during manufacturing, and that this secret information is unique to each device, unknown, and unpredictable to any person.

TE7.2 The tester shall examine, cite, and reference any supporting documentation submitted by the vendor. The documentation should detail that the device will be authenticated at the facility of initial deployment by means of secret information placed in the device during manufacturing. The secret information is installed in the device under dual control to ensure that it is not disclosed during installation, or the device may use an authenticated public-key method.

TE7.3 The tester shall examine a sample of documentation for secret information, including user documentation, and device-validation procedures to ensure procedures are followed.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TE7.4 The tester shall interview those responsible for the control processes—including engineers, software developers, line staff, supervisory staff, technical management, etc.—to verify that the approved and documented procedures are known and understood by all affected parties.

TE7.5 The tester shall observe the device secret installation and validation process to ensure that documented and approved procedures are followed subsequent to production. The tester shall verify that if secret information is placed in the device during manufacturing, then this secret information is unique to each device, unknown and unpredictable to any person, and installed in the device under dual control to ensure that it is not disclosed during installation.

DTR E8 Component Control during Design and Development

Security measures are taken during the development and maintenance of POI security-related components. The manufacturer must maintain development-security documentation describing all the physical, procedural, personnel, and other security measures that are necessary to protect the integrity of the design and implementation of the POI security-related components in their development environment. The development-security documentation shall provide evidence that these security measures are followed during the development and maintenance of the POI security-related components. The evidence shall justify that the security measures provide the necessary level of protection to maintain the integrity of the POI security-related components.

TE8.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail the design and development processes and procedures to ensure security measures are followed during the development and maintenance of the POI security-related components. Furthermore, the approved documentation must justify that the security measures provide the necessary level of protection to maintain the integrity of the POI security-related components.

TE8.2 The tester shall examine a sample of approved documentation for component control during design and development, including user documentation, and component validation procedures to ensure procedures are followed.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TE8.3 The tester shall interview those responsible for component control processes—including engineers and line staff, supervisory staff, technical management, etc.—to verify that the approved documented procedures are known and understood by all affected parties.

TE8.4 The tester shall observe the approved component control procedures to ensure security measures are followed during the development and maintenance of the POI security-related components.

TE8.5 The tester shall verify that the manufacturer maintains approved development-security documentation describing all the physical, procedural, personnel, and other security measures that are necessary to protect the integrity of the design and implementation of the POI security-related components in their development environment and that this provides evidence that these security measures are followed during the development and maintenance of the POI security-related components.

DTR E9 Repair and Inspection Control

Controls exist over the repair process at all POI vendor-authorized repair facilities, including the resetting of tamper mechanisms, and the inspection/testing process subsequent to repair to ensure that the device has not been subject to unauthorized modification.

Guidance

Completed devices and any of their components must be controlled and stored in tamper-evident packaging and/or stored in an accessed controlled area until shipped. The device or components should be checked prior to shipping to ensure the device has not been modified or tampered.

Device undergoing repair must be in a dual-access-controlled area or decommissioned such that all sensitive information and CSPs are cleared from the device prior to being returned to the device manufacturer.

TE9.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail repair, including the resetting of tamper mechanisms, inspection, and post-inspection control processes and procedures for storage and validation of devices or their components subsequent to repair and inspection but prior to shipping.

TE9.2 The tester shall examine, cite, and reference any supporting documentation submitted by the vendor. The approved documentation shall detail:

- The inspection process and tamper-evident packaging or the access-controlled area where devices and components are stored prior to shipping.
- For when a device is returned to the vendor for maintenance, mechanisms are in place to automatically cause the erasure of all previously loaded acquirer secret keys upon servicing the device—e.g., loading a new public RSA key causes the erasure of all previously loaded secret keys.

TE9.3 The tester shall examine a sample of approved documentation for repair, including the resetting of tamper mechanisms, inspection and post-inspection storage, including inventory logs, repair logs, shipping documentation, and device-validation procedures to ensure procedures are followed.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TE9.4 The tester shall interview those responsible for repair, inspection, and post-inspection control processes—including engineers, software developers, line staff, supervisory staff, technical management, etc.—to verify that the approved and documented procedures are known and understood by all affected parties.

TE9.5 The tester shall observe the device repair, inspection, and post-inspection storage process to ensure that procedures are followed subsequent to production.

TE9.6 The tester shall examine the vendor’s tamper-evident packing or accessed-controlled area to ensure unauthorized access to devices or their components is not possible.

DTR E10 Vendor Vulnerability Assessment Procedures

The device vendor has internal policies and procedures that ensure that the vendor maintains an effective process for detecting vulnerabilities that may exist within its device. This process is expected to be robust enough to include all interfaces defined in Requirement D1 and to detect vulnerabilities which may have not been publicly known during the last vulnerability assessment.

Guidance

The intent of this requirement is to ensure that the vendor has an effective process for detecting vulnerabilities within the firmware.

It is understood that vulnerability survey only represents a snapshot in time, and that vulnerabilities may become known in the public domain after that time. It is therefore expected that the vulnerability analysis incorporate up-to-date assessment mechanisms.

The mechanism the vendor uses to assess each protocol and interface should be effective for that protocol or interface. For example, it is not acceptable to use automated network scanning tools to assess a client protocol for vulnerabilities.

If an OEM module is present and shares resources with the rest of the device, a vulnerability assessment is required to ensure that the OEM module cannot adversely impact the function of the device.

The scope of the vulnerability assessment includes all firmware executing inside the device that may affect security.

TE10.1 The tester shall examine any relevant documentation distributed by the vendor to ensure that the vendor has provided for effective detection of vulnerabilities in each of the protocols and interfaces defined in D1. The tester shall list the methods and tools used by the vendor for vulnerability assessment including but not limited to test scripts, practical tests, review of information in public domain, and security analysis to locate vulnerabilities.

TE10.2 The tester shall examine the vendor’s documentation and verify that a process for the classification of newly detected vulnerabilities exists. The classification includes an analysis of the effect on the device, correct description, a level of criticality, and mitigation measures for each vulnerability.

TE10.3 The tester shall examine the vendor’s documentation for quality assurance and testing to verify that it contains analysis including methods such as but not limited to; test scripts, practical tests, review of information in public domain, and security analysis to locate vulnerabilities. The tester shall verify that these are effective mechanisms for detecting.

TE10.4 The tester shall examine the vendor’s documentation to verify that a process to create an auditable record of completed vulnerability assessments exists. The record is to include what testing has been done, the results of the assessment, and sign-off from management.

This vulnerability assessment cannot be older than three months since the beginning of the security evaluation.

TE10.5 The tester shall examine the vendor’s documentation to verify that the reports are reviewed internally, and if vulnerabilities are detected the proper steps are taken.

DTR E11 Vulnerability Assessment of all Interfaces

The device has undergone a vulnerability assessment to ensure that the protocols and interfaces listed in D1 do not contain exploitable vulnerabilities.

a) The vulnerability assessment is supported by a documented analysis describing the security of the protocols and interfaces.

b) The vulnerability assessment is supported by a vulnerability survey of information available in the public domain.

c) The vulnerability assessment is supported by testing.

Guidance

The intent of this requirement is to ensure that the vulnerability assessment process evaluated under Requirement E10 has been followed.

Review the vulnerability assessment for each interface as defined in D1.

Each interface shall be assessed and reported separately.

If the OEM module shares resources with the rest of the device, a vulnerability assessment is required to ensure that the OEM module cannot adversely impact the function of the device.

TE11.1 The tester shall examine relevant documentation, such as scan reports, test reports or vulnerability analysis documentation submitted by the vendor to verify that it supports the vendor responses.

TE11.2 The tester shall examine the vendor's sign-off form specified in Requirement E10 to ensure that it has been completed and signed off by management.

TE11.3 The tester shall examine any anomalies indicated by the vendor and determine whether they are correctly described, have a level of criticality assigned, and that mitigation measures are suggested.

DTR E12 Vulnerability Disclosure

The device vendor has vulnerability disclosure measures in place for the device.

a) The vulnerability-disclosure measures are documented.

b) The vulnerability-disclosure measures ensure a timely distribution of information about newly found vulnerabilities. This information includes identification, description, and assessment of the vulnerabilities.

c) The vulnerability-disclosure measures ensure a timely distribution of mitigation measures.

Guidance

The intent of this requirement is to ensure that the vendor has a process whereby it is able to disclose to its customers vulnerabilities that affect the device.

The vendor must maintain documentation that describes their internal processes.

The vulnerability disclosure process applies to all code released by the device vendor.

TE12.1 The tester shall examine relevant documentation, such as a user guide or the merchant implementation guide submitted by the vendor, to verify that it supports the vendor responses.

TE12.2 The tester shall examine the documentation to verify both the timely creation of mitigation measures for newly found vulnerabilities and that procedures exist to continually update and document all vulnerabilities.

TE12.3 The tester shall perform any additional tests necessary to validate the vendor's vulnerability-disclosure measures including auditing the vendor's vulnerability management program output. The tester should use his or her own judgment in determining the appropriate tests and shall document why the test evidence and methods used are valid. The vendor shall provide any necessary test support to access and use the interfaces under test.

F – Between Manufacturer and Facility of Initial Key Loading or Facility of Initial Deployment

Compliance can be proven by evidence that the procedures comply by design via:

- Evidence of existence—i.e., that these procedures are implemented. For example, the lab has received procedures from the company that implements the requirement.
- When these procedures are in use, samples of the process in operation. For this purpose—in a timeframe of 3, 6, or 12 months—samples can be randomly collected and validated. Example: For F3, the lab can randomly select a few occasions and ask for the related log belonging to the received shipments proving that the packaging is validated on tamper evidence.

DTR F1 Shipping Tamper-Protection Documentation

The POI should be protected from unauthorized modification with tamper-detection security features, and customers shall be provided with documentation (both shipped with the product and available securely online) that provides instruction on validating the authenticity and integrity of the POI.

Where this is not possible, the POI is shipped from the manufacturer's facility to the initial key-loading facility or to the facility of initial deployment and stored enroute under auditable controls that can account for the location of every POI at every point in time—such as the use of serialized tamper-evident packing for all devices with no tamper detection, in conjunction with thorough physical inspection (possibly including sampling of HW internals) upon reception.

Where multiple parties are involved in organizing the shipping, it is the responsibility of each party to ensure that the shipping and storage they are managing is compliant with this requirement. In the absence of defined agreements stipulating otherwise, the POI vendor remains responsible.

Guidance

The product shall be protected for at all points during the shipping process. Tamper-detection security features, instructions for receiving and inspection, and documentation should tamper be suspected, must be provided and used.

Prior to initial financial-key loading, the POI should be protected against modification. Such protection shall be a combination of the characteristics of the device—i.e., tamper evidence, resistance, and responsiveness—and device-management procedures. If the device has any manufacturer keys loaded, compromise shall be both prevented and detected.

TF1.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail the shipping process, tamper protection, instructions for receiving and inspection, as well as procedures for suspected tamper evidence for customers that provides instruction on validation of the authenticity and integrity of the POI. Alternatively, the approved documentation must detail how the POI is shipped from the manufacturer's facility to the initial key-loading facility or to the facility of initial deployment and stored enroute under auditable controls that can account for the location of every POI at every point in time.

TF1.2 The tester shall examine approved documentation detailing that where multiple parties are involved in organizing the shipping, it is the responsibility of each party to ensure that the shipping and storage they are managing is compliant with this requirement. In the absence of defined agreements stipulating otherwise, the POI vendor remains responsible.

TF1.3 The tester shall examine a sample of documentation for tamper protection, inspection, and transfer documentation, including inventory logs, shipping documentation, and device-validation procedures to ensure procedures are followed and describe how this shows compliance to this requirement.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TF1.4 The tester shall interview those responsible for the shipping control processes—including engineers, software developers, line staff, supervisory staff, technical management, etc.—to verify that the approved documented procedures are known and understood by all affected parties.

TF1.5 The tester shall observe the shipping process to ensure that customers are provided documentation (both shipped with the product and available securely online) that provides instruction on validating the authenticity and integrity of the POI or alternatively, the POI is shipped from the manufacturer's facility to the initial key-loading facility or to the facility of initial deployment and stored enroute under auditable controls.

DTR F2 Device-Accountability Transfer Procedures

Procedures are in place to transfer accountability for the device from the manufacturer to the facility of initial deployment. Where the device is shipped via intermediaries such as resellers, accountability will be with the intermediary from the time at which they receive the device until the time it is received by the next intermediary or the point of initial deployment. In the absence of defined agreements stipulating otherwise, the POI vendor remains responsible.

Guidance

The product shall be accounted for at all points during the shipping process until transfer of responsibility. Documentation shall be maintained for each POI.

TF2.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail the instructions for receiving and inspection, as well as procedures for the transfer of responsibility. Furthermore, the documentation should detail that where the device is shipped via intermediaries such as resellers, accountability will be with the intermediary from the time at which they receive the device until the time it is received by the next intermediary or the point of initial deployment. In the absence of defined agreements stipulating otherwise, the POI vendor remains responsible.

TF2.2 The tester shall examine a sample of transfer documentation, including inventory logs, and shipping documentation, to ensure procedures are followed.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TF2.3 The tester shall interview those responsible for the shipping control processes—including engineers, software developers, line staff, supervisory staff, technical management, etc.—to verify that the approved documented procedures are known and understood by all affected parties.

TF2.4 The tester shall observe of the shipping and transfer procedures to ensure that procedures are followed to transfer accountability for the device from the manufacturer to the facility of initial deployment.

TF2.5 The tester shall verify that where the device is shipped via intermediaries such as resellers, accountability is with the intermediary from the time at which they receive the device until the time it is received by the next intermediary or the point of initial deployment.

DTR F3 Shipping Security Procedures

While in transit from the manufacturer's facility to the initial key-loading facility, the device is shipped and stored containing a secret that:

- ***Is immediately and automatically erased if any physical or functional alteration to the device is attempted, and***
- ***Can be verified by the initial key-loading facility but cannot feasibly be determined by unauthorized personnel.***

TF3.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail the shipping process, tamper protection, instructions for receiving and inspection, as well as procedures for suspected tamper evidence and describe how this shows compliance to this requirement.

TF3.2 The tester shall examine a sample of documentation for tamper protection, inspection and transfer documentation, including inventory logs, shipping documentation, and device-validation procedures to ensure procedures are followed.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TF3.3 The tester shall interview those responsible for the shipping control processes—including engineers, software developers, line staff, supervisory staff, technical management, etc.—to verify that the approved documented procedures are known and understood by all affected parties.

TF3.4 The tester shall examine a sample of documentation for tamper protection using secret keys to ensure approved procedures for validation at the key-loading faculty are followed.

TF3.5 The tester shall observe the shipping process to ensure that POI devices are shipped and stored containing a secret that is immediately and automatically erased if any physical or functional alteration to the device is attempted, that can be verified by the initial key-loading facility, but that cannot feasibly be determined by unauthorized personnel.

DTR F4 Development-Security Documentation

The device's development-security documentation must provide means to the initial key-loading facility to assure the authenticity of the device's security-relevant components.

Guidance

The means to verify the authenticity of the device may include tools such as an SCD.

TF4.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail the device's development-security documentation to ensure the authenticity of the device's security-relevant components and describe how this shows compliance to this requirement.

TF4.2 The tester shall examine a sample of documentation for device-development security, including user documentation, and component validation procedures to ensure procedures are followed.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TF4.3 The tester shall interview those responsible for the initial key-loading facility—including engineers and line staff, supervisory staff, technical management, etc.—to verify that the approved documented procedures are known and understood by all affected parties.

TF4.4 The tester shall observe the secure development component control procedures to ensure security measures are followed during the initial key loading.

DTR F5 Authenticity of POI Security-Related Components

If the manufacturer is in charge of initial key loading, the manufacturer must verify the authenticity of the POI security-related components.

TF5.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail procedures for the manufacturer to ensure the authenticity of the device's security relevant components.

TF5.2 The tester shall examine a sample of documentation, including user documentation, and component validation procedures to ensure procedures are followed.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TF5.3 The tester shall interview those responsible for the initial key-loading facility—including engineers and line staff, supervisory staff, technical management, etc.—to verify that the approved and documented procedures are known and understood by all affected parties.

TF5.4 The tester shall observe the initial key-loading procedures to ensure the authenticity of the POI security-related components is verified.

DTR F6 Authenticity of POI Security-Related Components for Key-Loading Facility

If the manufacturer is not in charge of initial key loading, the manufacturer must provide the means to the initial key-loading facility to assure the verification of the authenticity of the POI security-related components.

TF6.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail procedures provided by the manufacturer to the initial key-loading facility to assure the authenticity of the device’s security-relevant components for the initial key-loading facility.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TF6.2 The tester shall interview those responsible for the initial key-loading facility—including engineers and line staff, supervisory staff, technical management, etc.—to verify that the approved and documented procedures are known and understood by all affected parties.

TF6.3 The tester shall examine a sample of documentation, including user documentation and component-validation procedures to ensure procedures are followed at the initial key-loading facility

DTR F7 Unique Visible Identifier

Each device shall have a unique visible identifier—i.e., model name and hardware version—affixed to it. This information shall also be retrievable by a query.

TF7.1 The tester shall examine and cite any supporting documentation submitted by the vendor. The documentation should detail how the change-control procedures required in E1 are used in compliance with this requirement.

TF7.2 The tester shall verify and show evidence that the model name and hardware version are retrievable from the device by a query.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TF7.3 The tester shall observe the manufacturing process to ensure the visible identifier is affixed to each device.

TF7.4 The tester shall sample the devices to ensure that each visible identifier is unique and is consistent with the identifiers checked with the change-control procedure in as required in E1.

DTR F8 Operational Management Manual

The vendor must maintain a manual that provides instructions for the operational management of the POI. This includes instructions for recording the entire life cycle of the POI security-related components and of the manner in which those components are integrated into a single POI, e.g.:

- ***Data on production and personalization***
- ***Physical/chronological whereabouts***
- ***Repair and maintenance***
- ***Removal from operation***
- ***Loss or theft***

Guidance

The operational management manual provides instructions and information concerning the identification, use, repair, updates, and configuration of hardware and firmware components of the device. This manual is in addition to user and application operation and configuration manuals.

TF8.1 The tester shall examine and cite any supporting documentation submitted by the vendor and describe how this shows compliance to this requirement. The documentation should include instructions for recording the entire life cycle of the POI security-related components and the manner in which those components are integrated into a single POI. The operations management manual must be current and up to date.

Where required to validate via site inspection, the tester shall complete the following additional steps:

TF8.2 The tester shall interview those responsible for maintaining the operation management manual—including engineers and software developers, supervisory staff, technical management, etc.—to verify that the approved and documented procedures are known and understood by all affected parties.

TF8.3 The tester shall examine a sample of the operations management manual to ensure procedures are followed and recorded.

Appendix A: Criteria for the Privacy Screen Design

A.1.1 Upright (for example, Unattended) Privacy Screen Design Criteria to be met by the Device's Design

The following are examples of device privacy screens being provided by the device itself that are compliant with *PCI PTS POI Security Requirements*. Other designs may also be acceptable.

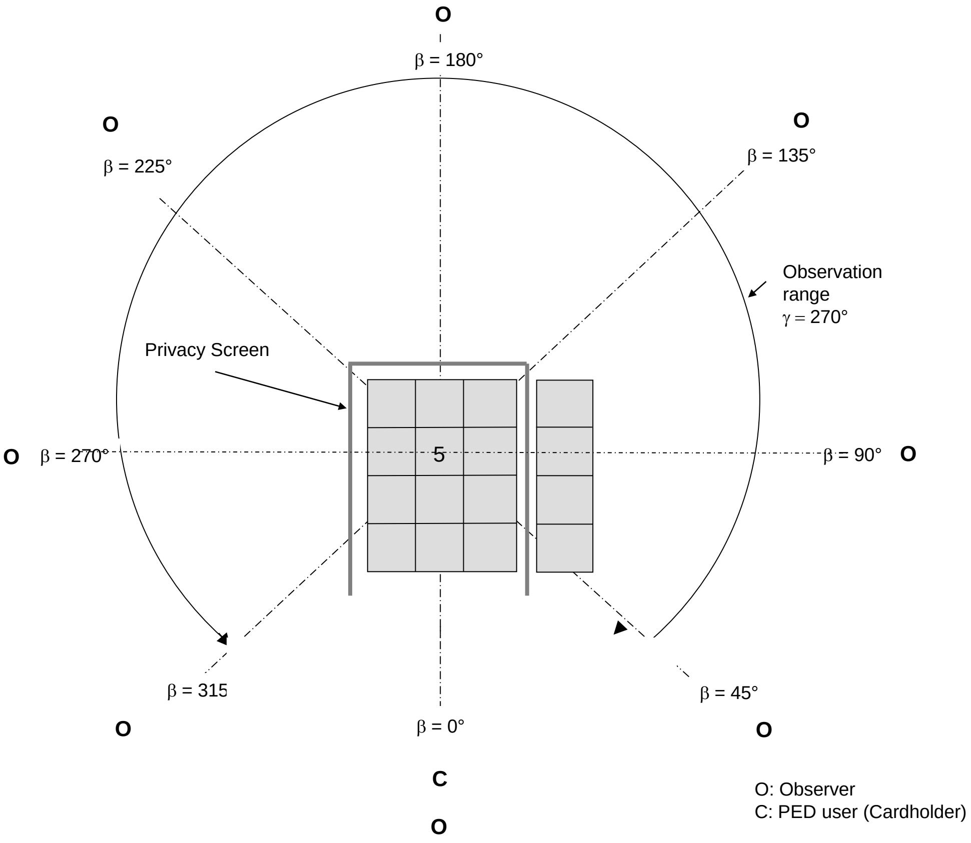


Figure A1: Sample device with privacy screen range, bird's eye view

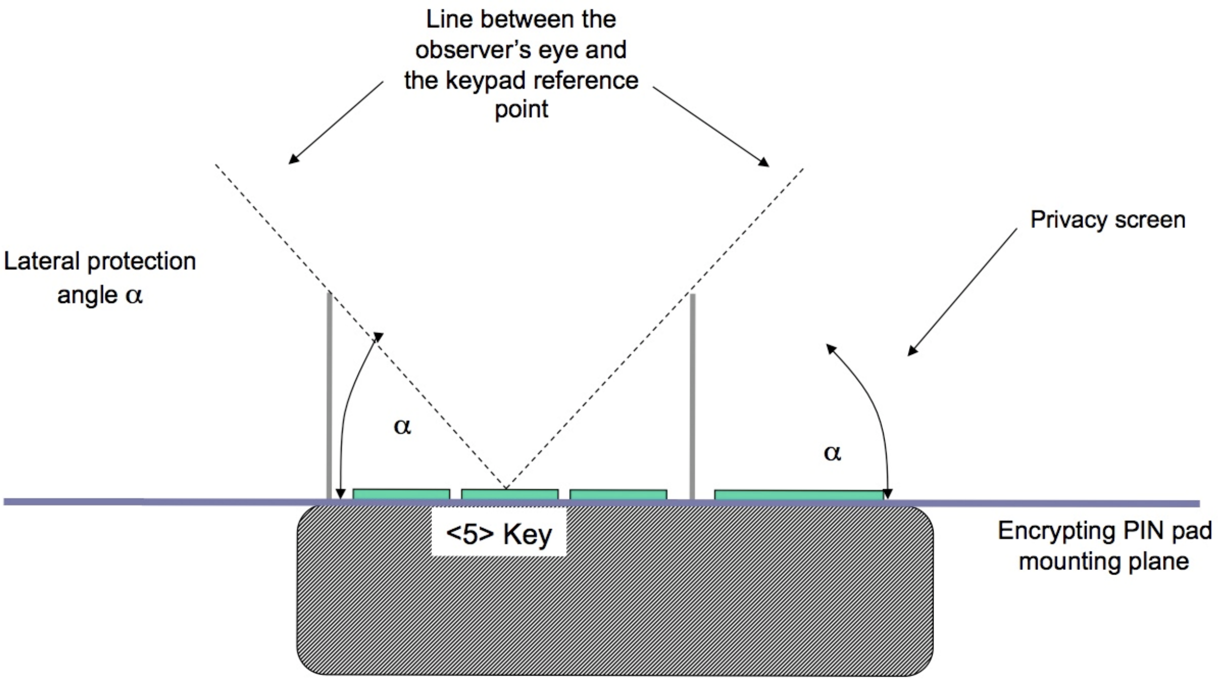

Line between the observer's eye and the keypad reference point
Privacy screen
Lateral protection angle α
α
α
<5> Key
Encrypting PIN pad mounting plane

Figure A2: Sample device keypad, sectional drawing from the "0" side

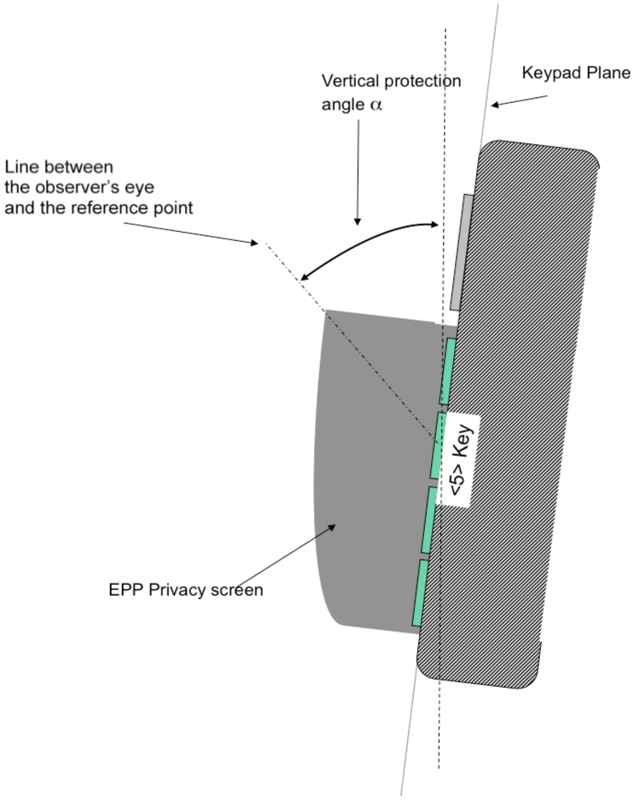

Vertical protection angle α
Keypad Plane
Line between the observer's eye and the reference point
<5> Key
EPP Privacy screen

Figure A3: Sample device keypad, side view

The angles in the figures above are defined as follows:

α: Angle between the vertical plane through the “5” key and a virtual line which connects the “5” key and an observer’s eye

β: Horizontal position of an observer relative to the PIN entry device’s position

γ: Horizontal range which is to be covered by the privacy screen

δ: Angle between the keypad plane and the horizontal plane

Design rules:

1. These definitions apply to a privacy shield, which is provided as design property by the device. It may be a part of the PIN entry device or provided by the device’s cabinet. The rules and the figures above are to be considered as guidelines, which may be replaced by other means of at least the same efficiency.

2. The keypad reference point is taken as the column position in the middle of the keypad in the row containing the numeric key “5.”

3. The privacy screen of the device is to be placed horizontally or slightly tilted (0 ≤ **δ** ≤ 45°) and shall provide the following protection angles:

Horizontal angle β	Remark	Vertical angle α
315° ≤ **β** ≤ 45°:	Within this range of β the cardholder deters an observer with her/his body.	N/A
45° ≤ **β** ≤ 90° 270° ≤ **β** ≤ 315°:	Within these ranges visual observation of the keypad is partially blocked by the cardholder. The protection angle α shall be at least 35°. Please note that the front end of the privacy screen must be higher if the device is tilted.	**α** ≥ 35°
90° ≤ **β** ≤ 270°:	The protection angle shall be at least 40°. The display side of the privacy screen may be lowered as the device is tilted against the horizontal plane.	**α** ≥ 40°

The vertical angles given in the table above are with respect to the horizontal plane (see figure above). If by design of the device the keypad is tilted toward the cardholder, the backside of the privacy screen may be lower.

4. If the device is to be placed vertically or tilted by 45° or more, the requirements under Step 3 will apply accordingly, using the vertical plane instead of the horizontal plane as the reference for the angle **α**.

5. The protection is based on viewing angles and does not imply a specific technical implementation like physical shields. If the keypad is implemented as a touchscreen, the viewing barrier may be implemented by polarizers (for example, as film embedded within layers of a touchscreen), which deter the observation from the sides. The up (clerk) side must be implemented as a physical shield.

A.1.2 Countertop (for example, Attended Device) Privacy Screen Design Criteria to be met by the Device's Design

The following are examples of device's privacy screens being an integral part of the device that are compliant with *PCI PTS POI Security Requirements*. Other designs may also be acceptable.

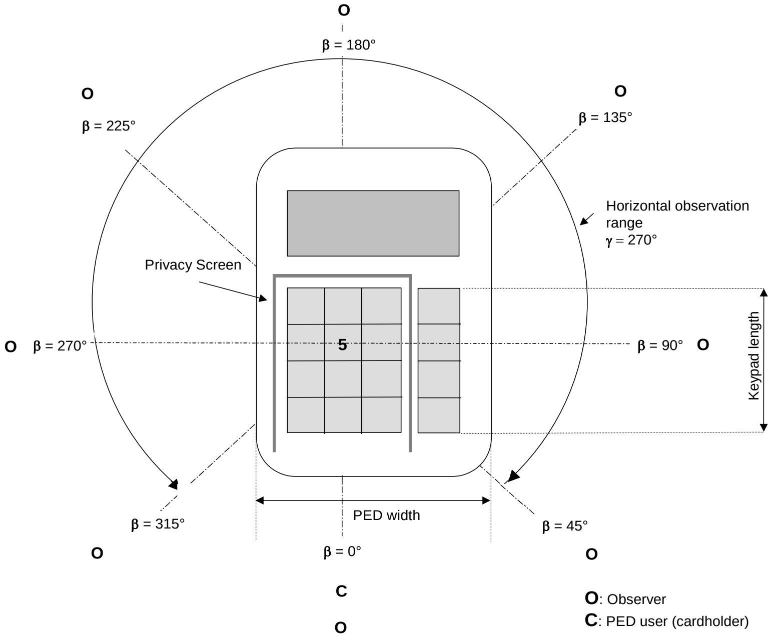


Figure A4: Sample device with privacy screen range, bird's eye view

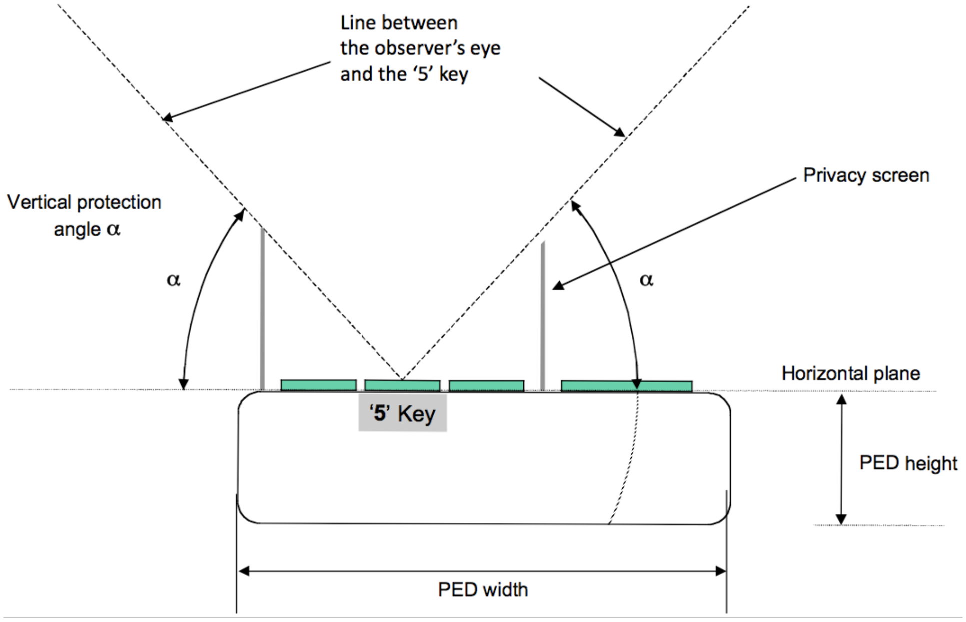

Line between the observer's eye and the '5' key
Privacy screen
Vertical protection angle α
α
α
Horizontal plane
'5' Key
PED height
PED width

Figure A5: Sample device, front side view

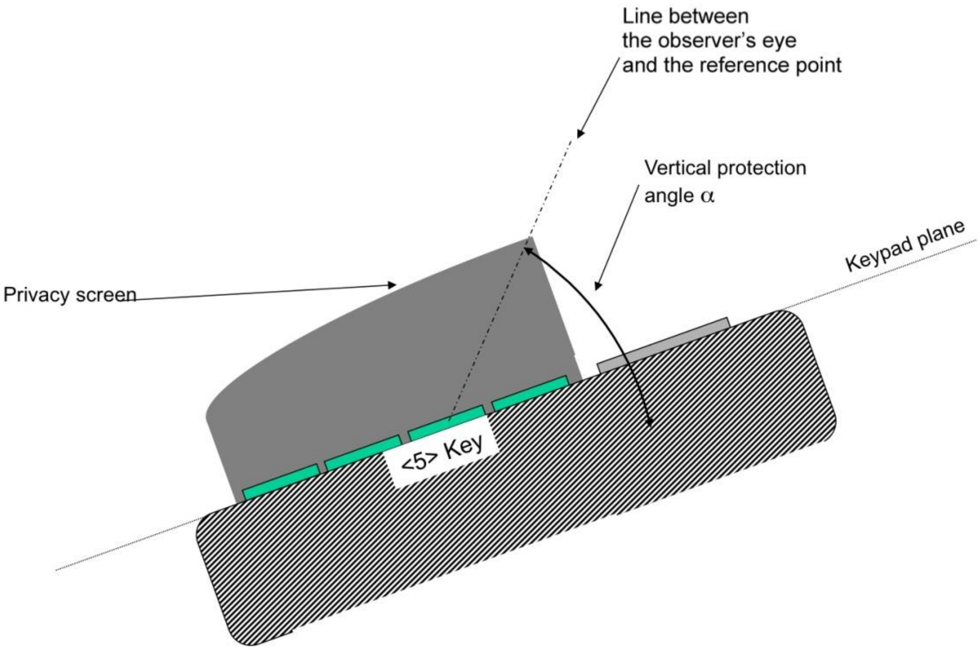


Figure A6: Sample device, side view

The angles in the figures above are defined as follows:

α: Angle between the horizontal plane through the "5" key and a virtual line which connects the "5" key and an observer's eye

β: Horizontal position of an observer relative to the PIN entry device user's position

γ: Horizontal range which is to be covered by the privacy screen

δ: Angle between the keypad plane and the horizontal plane

Design rules:

1. The requirements differentiate between an attended device, handheld, touchscreen, or an unattended device. The intended use of the device must be clearly stated.

   The privacy screen of the device is to be placed horizontally or slightly tilted (0 ≤ δ ≤ 45°) and shall provide the following protection angles:

Horizontal angle β	Remark	Vertical angle α
315° ≤ β ≤ 45°:	Within this range of β the cardholder deters an observer with her/his body.	N/A
45° ≤ β ≤ 90° 270° ≤ β ≤ 315°:	Within these ranges visual observation of the keypad is partially blocked by the cardholder. The protection angle α shall be at least 35°. Please note that the front end of the privacy screen must be higher if the PIN entry device is tilted.	α ≥ 35°
90° ≤ β ≤ 270°:	The protection angle shall be at least 40°. The display side of the privacy screen may be lowered as the PIN entry device is tilted against the horizontal plane.	α ≥ 40°

   The vertical angles given in the table above are with respect to the horizontal plane (see figure above). If by design of the PIN entry device the keypad is tilted toward the cardholder, the backside of the privacy screen may be lower.

2. If the device is to be placed vertically or tilted by 45° and more, the requirements under Step 3 will apply accordingly, using the vertical plane instead of the horizontal plane as the reference for the angle α.

3. The protection is based on viewing angles and does not imply a specific technical implementation like physical shields. If the keypad is implemented as a touchscreen, the viewing barrier may be implemented by polarizers (for example, as film embedded within layers of a touchscreen surface), which deter the observation from the sides. The up (clerk) side must be implemented as a physical shield.

4. **A handheld device** must by weight, size, and shape encourage its handheld operation. The criteria for a device with a physical keypad are:

   a) Weight should be 500 grams or less;

   b) Width at the “5” key should not be more than three (3) inches or 7.62 cm;

   c) Sum of width and height at the “5” key should not be more than four (4) inches or 10.16 cm; and

   d) Keypad length should not be more than four (4) inches or 10.16 cm.

   If the device’s properties clearly fall outside these ranges, it will not be accepted as a handheld device for purposes of this test. A handheld device must by its size and case shape encourage its handheld use. Being small may not be sufficient.

For devices with a touchscreen, the criteria are similar to that of tablets and smartphones:

a) Weight should be 600 grams or less;

b) Width of the virtual effective keypad at the “5” key should not be more than three (3) inches or 7.62 cm;

c) Length of the virtual effective keypad should not be more than four (4) inches or 10.16 cm; and

d) The diagonal of the display should be not more than ten (10) inches or 25.4 cm.

However, the guidelines listed are suggestions, not requirements.

A.2 Privacy Screen Design Criteria to be met by the Device's Installed Environment

The following techniques can be employed to provide for effective screening of the PIN-entry keypad during the PIN entry process. These methods would typically be used in combination, though in some cases a method might be used singly.

Note: *This option does not preclude the use of privacy mechanisms as defined in A1 but allows less restrictive physical mechanisms.*

- Positioning of terminal on the check-stand in such way as to make visual observation of the PIN-entry process infeasible. Examples include:
  - Visual shields designed into the check-stand. The shields may be solely for shielding purposes or may be part of the general check-stand design, for example, used as selling area.
  - Position the device so that it is angled in such a way to make PIN spying difficult.
- Pop-up (temporary) privacy shield attached to the device-mounting stand. Consumer (through education and prompting) or merchant would put the shield in place during PIN entry
- Installing device on an adjustable stand that allows consumers to swivel the terminal sideways and/or tilt it forwards/backwards to a position that makes visual observation of the PIN-entry process difficult.
- Positioning of in-store security cameras such that the PIN-entry keypad is not visible.
- Instructing the cardholder regarding safe PIN-entry. This can be done with a combination of
  - Signage on the device;
  - Prompts on the display, possibly with a "click-through" screen;
  - Potentially, literature at the point of sale; and
  - A logo for safe PIN-entry process.

Other methods are possible as well. The above are examples of some of the methods a vendor can propose to protect PINs during PIN entry. The vendor must provide adequate techniques in the device documentation and also include a matrix showing which techniques should be used to protect against specific observation corridors. Table A1 on the following page shows an example matrix .

Table A1: Sample Matrix of Observation Corridors and PIN Protection Methods

Method	Observation Corridors[1]				
	Cashier	Customers in Queue	Customers Elsewhere	On-Site Cameras	Remote Cameras
Device Stand A	M	H	L	L	L
Device Stand B	H	H	H	L	M
Check-Stand A	L	M	M	L	H
Check-Stand B	H	H	M	H	H
Customer Instruction[2]	H	H	H	H	H

The matrix must show the purchaser of the device, the types of methods they may use to protect their customers' PINs. The appropriate methods would be selected in order to ensure an appropriate level of protection from all observation corridors.

[1] **L** = low, **M** = medium, **H** = high.

[2] Customer Instruction methods are less repeatable and therefore should be used in combination with other methods.

Appendix B: Physical Attack Potential Formula (Adopted from JIL)

Calculating Attack Potentials

This section presents factors that determine attack potentials and provides guidelines to help remove some of the subjectivity from this aspect of the evaluation process. Attack potential calculations in reports must accurately reflect the actual evaluated device's properties, coherently with this guidance. The approaches here should be adopted unless the tester determines that it would be inappropriate, in which case a strong rationale is required to justify the validity of the alternative approach. The examples shown here are generic descriptions and should not be assumed to apply to any specific device. Strong and detailed justification must be made for attributing high levels in ratings.

Identification and Exploitation

For an attacker wanting to exploit a vulnerability, the vulnerability must first be identified. This may appear to be a trivial separation, but it is an important one. To illustrate this, first consider a vulnerability that is uncovered following months of analysis by an expert, and a simple attack method published on the Internet. Compare this to a vulnerability that is well known but requires enormous expenditure of time and resources to exploit. Of course, factors such as time need to be treated differently in these cases.

In this document, "exploitation" and "initial exploitation" are alternatively used to designate "initial exploitation."

Factors to be Considered

The following factors should be considered for the analysis of the attack potentials required to exploit a vulnerability:

1. **Identification**

   a) Attack time for the various levels of expertise;

   b) Potential to acquire the required knowledge of the POI device's design and operation;

   c) Potential for the access to the POI device;

   d) Equipment required such as instruments, components, IT hardware, and software required for the analysis;

   e) POI device specific spare components.

2. **(Initial) Exploitation**

   a) Attack time for the various levels of expertise;

   b) Potential to acquire the required knowledge of the POI device's design and operation;

   c) Potential for the access to the POI device;

   d) Equipment required such as instruments, components, IT hardware, software required for the analysis;

   e) POI device specific spare components.

In many cases these factors don't depend on each other but might be substituted for each other in varying degrees. For example, expertise or hardware/software can be a substitute for time. A discussion of these factors follows. In most situations, the first identification phase shall include a complete simulation of the second phase—i.e., the initial exploitation phase. Therefore, it is not acceptable to allocate calculation factors (such as time, experience, equipment) of the simulation to an exploitation phase when these can be attributed to the identification phase.

Attack Time

The **attack time** is given in the time in hours taken by an attacker to identify or exploit an attack. If the attack consists of several steps, the attack time can be determined and added to achieve a total attack time for each of these steps. Actual labor time must be used instead of time expired as long as there is not a minimum attack time enforced by the attack method applied (for instance, the time needed for performing a side-channel analysis, data collections, or the time needed for an epoxy to harden).

In those cases where attendance is not required during part of the attack time, the attack time is to be taken as expired time divided by 3.

Expertise

Expertise refers to the level of generic capabilities including but not limited to knowledge, skills, experience, etc. of the application area or product type (for example, UNIX operation systems, and Internet protocols). Identified levels are as follows:

a) **Layman** is a person without professional or specialized knowledge in a particular subject. They are unknowledgeable compared to experts, proficient, or skilled persons, with no particular expertise but capable of implementing simple attack steps, or capable of implementing more complex attack steps under direction. For the purpose of exploitation, they can implement an attack based on a script or a written procedure, without requiring any particular skill.

b) **Skilled** persons are able to perform more complex tasks and attack steps without direction. They have the ability and training to perform a specific task well.

c) **Proficient** persons are highly competent and have the necessary ability, knowledge, and skill to perform complex attacks successfully. They are familiar with the security functionalities and behavior of the product. For the purposes of exploitation, proficient persons qualify when specific skills are required to conduct an attack successfully. Proficient personnel can acquire capabilities equivalent to the skill sets of PTS lab evaluators.

d) **Experts** are extremely knowledgeable and skillful in one or more areas. They are very familiar with the underlying algorithms, protocols, hardware components, physical and logical architectures, etc., implemented in the device or system type and the principles and concepts of security employed. Experts are capable of quickly devising new attack paths that require specific competencies similar to those of experienced PTS lab personnel.

If proficient expertise in various areas of technology is required for an attack—for example, on electrical engineering and cryptography—an expert level of expertise can be assumed. In most attack scenarios, the exploitation level of expertise is less than the required identification level of expertise.

Knowledge of the POI Device

Knowledge of the POI device refers to obtaining specific expertise in relation to the POI device. This is different from generic expertise but not unrelated to it. Identified levels are as follows:

a) **Public information** about the POI device (or no information): Information is considered public if it can be easily obtained by anyone (for example, from the Internet) or if it is provided by the vendor without any control mechanism—e.g., to any customer. Examples include open protocol specifications and electronic component datasheets. Information with automated access-control mechanisms (such as online account subscription) without human intervention classifies as Public.

b) **Restricted information** concerning the POI device (for example, as gained from vendor technical specifications): Information is considered restricted if it is distributed on request and the distribution is registered (for example, like the *PCI PTS POI DTRs*). Restricted information is distributed upon request and is subject to human-based control mechanisms. Examples of restricted information are mechanical drawings for OEM device integration, external command API specifications, partial Gerber files, and secure processor datasheets available under NDA.

c) **Sensitive information** about the POI device (for example, knowledge of internal design, which may have to be obtained by "social engineering" or exhaustive reverse-engineering). Sensitive information is not intended to be distributed to external entities and is obtained by means such as "social engineering" theft or coercion. Typical examples are terminal schematics and firmware source code.

Distinction must be made between information required to identify the vulnerability and the information required to exploit it, especially in the area of sensitive information. Requiring restricted or sensitive information for exploitation would be unusual.

Access to the POI Device

Access to the POI device is also an important factor. It is assumed here that the POI device would be purchased or otherwise obtained by the attacker and that beside other factors there is not any time limit in analyzing or modifying the POI device. Differences are defined in the status and functionality of the device to be analyzed/tested.

a) **Mechanical samples** are non-functional and are used merely to study the mechanical design or for supplying spare parts.

b) **Functional samples without working keys** might be used for the logical and electrical behavior of the device but are not loaded with working keys and are therefore not functional within a payment network or with real payment cards. Such devices might be regularly purchased. Please note that these also include devices fitted with test or dummy keys.

c) **Functional samples with working keys** are fully functional devices, which might be used to verify an attack method or to actually perform an attack.

If more than one sample is needed in any category, instead of multiplying the points by the number of samples, the following factors must be used:

Table B1: Multiple Samples Factors

Number of Devices	Factor
1	1
2	1.5
3–4	2
5–10	4
> 10	5

The requirement for multiple devices during either the identification or the exploitation phase of an attack value calculation depends upon the difficulty of attacking a device, and the risk that the device may be tampered during the attack. However, PCI expects that most attacks can be performed with only one or two samples in the identification phase, and a single sample in the exploitation phase. Strong justification explaining why multiple sample devices are necessary must be provided when such additional samples are necessary to meet the minimum attack potential.

Equipment

Equipment refers to the equipment that is required to identify or exploit vulnerability. See Appendix C for details of equipment classification.

a) **Standard equipment** is equipment that is readily available to the attacker, either for the identification of vulnerability or for an attack. This equipment can be readily obtained—for example, from retail outlets or acquired/downloaded from the Internet.

b) **Specialized equipment** isn't readily available to the attacker but could be acquired without undue effort.

c) **Bespoke equipment** is not readily available to the public, as it might need to be specially produced (for example, very sophisticated software), or because the equipment is so specialized that its distribution is controlled (for example, X-ray equipment), possibly even restricted. Bespoke equipment that can be rented must be treated as specialized equipment. Software that has been developed during the identification phase is considered as bespoke equipment and must not additionally be considered in the exploitation phase.

d) **Chip-level equipment**, necessary to implement chip-level attacks, is generally not widely available on the market and its access is restricted. It is also very expensive, and therefore is considered as a category on its own. Only the following equipment belong to this category:

   - Focused Ion Beam
   - Scanning Electron Microscope

If, for one phase (identification or exploitation), equipment from different levels is required, only the highest level shall be retained. Hence, only the highest level of equipment shall be counted in any one phase of an attack calculation.

Specialist expertise and equipment are concerned with there being an implicit relationship between an attacker's expertise and the ability to effectively make use of equipment in an attack. The weaker the attacker's expertise, the lower the potential to effectively use equipment. Likewise, the greater the expertise, the greater the potential for equipment to be used in the attack. Although implicit, this relationship between expertise and the use of equipment does not always apply—for instance, when environmental measures prevent an expert attacker's use of equipment; or when, through the efforts of others, attack tools requiring little expertise for effective use are created and freely distributed (for example, via the Internet).

Parts

Parts refer to components required to hide the signs of an attack; to otherwise replace components that have been broken during an attack, like a case part, a display or a printer; to create a data-monitoring or communicating bug; or are otherwise needed to perform the attack. PIN-disclosing bugs belong to this category.

a) **Standard parts** are readily available to the attacker, either by purchasing them from a supply store or by re-using parts from a mechanical sample of the same device.

b) **Specialized parts** are not readily available to the attacker but could be acquired without undue effort. These might be parts that are not readily available from a supply store but can be ordered from stock and require delivery time or a certain minimum component count for purchase.

c) **Bespoke parts** are not readily available and have to be specifically manufactured. It is very unlikely that an attack requires bespoke spare parts.

Software required for a PIN-disclosing bug is typically straightforward to implement and would not be considered bespoke. Bespoke software would be software that requires significant time and expertise to develop such as is required for side channel attacks. PCI requires strong justification to be provided when bespoke parts is indicated as necessary for an attack.

PIN bugs must often be customized for a specific device. Due to numerous possible variations in bug form, function, and complexity, PCI does not provide standard point values for PIN bugs. The evaluation lab is responsible for addressing this as part of the device evaluation. The development of an appropriate PIN-disclosing bug is to be included in the Identification calculation, as are other aspects of attack development.

Parts used during the Identification phase that can be used in the initial exploitation must be counted fully in the Exploitation phase to equalize the attack potential value across all exploitations. While equipment readily lends itself to reuse for each exploitation, parts are typically a one-time use for each exploitation. Each exploitation should have the same attack potential value. Accounting for parts that are reused in the initial exploitation only in the Identification phase, or even splitting between the Identification and Exploitation phases, will result in the initial exploitation having a lower attack-potential value than the actual subsequent exploitations. Therefore, parts used during the Identification phase that can be used in the initial exploitation must be counted fully in the Exploitation phase to equalize the attack-potential value across all exploitations. If it is not readily reusable (the part once used in installation becomes unusable for exploitation because, for example, it is glued with epoxy and difficult to remove), it can be accounted for twice—once in the Identification phase and again in the Exploitation phase.

Rating Exploitation

It is important to note that only initial exploitation is considered, as further exploitations can reuse equipment and knowledge and are subject to optimization, which cannot be easily assessed through laboratory evaluations. Attacks are based on an attack performed on a single device. If the attack potential is met, then the requirement is met regardless of whether or not applying the attack to additional devices is less than the attack potential.

The following items in calculation are typically subject to reuse in further exploitation phases:

- Equipment
- Knowledge

If several attack scenarios lead to the same potential in total, then the one that has the lowest cost in exploitation, exclusive of the items above, must be considered. Attack calculations must allocate ratings that assume the most conservative trade-offs between time, expertise and equipment. Particularly, attack calculations shall not distribute ratings in a way that increases overall and/or exploitation minimum ratings above the most conservative approach.

Attack Considerations

If an attack requires damage to a part of the POI casing, the attack does not need the replacement of that entire casing to hide the tamper evidence. This is because there are many ways in which tamper evidence can be hidden, either through the installed environment of the POI, through the application of stickers, epoxy putty and paint, or other methods. Complete replacement of the plastic part is to be considered only when it adds little or no cost to the overall attack—e.g., when the process causing the damage to the casing results in the disablement of the tamper detection, thereby facilitating the removal and replacement of the casing at no cost.

There isn't any limit to the level of damage to a POI that may be "covered up" without casing replacement. It is easy to either cut out a suitable part from a mechanical sample (obtained during identification) or 3D-print a part for the replacement area. Paint or stickers can easily hide any seams or damage as it is common in the marketplace for deployed POIs to have painted plastic casings or stickers of some type. Therefore "pure" casing damage, including full case replacement, will receive 1 point for a standard part in the exploitation phase.

Damage to the front or back should not be treated differently, even though there is increased visibility of frontal damage, unless the rear can be expected to not be visible—e.g., in an EPP—where any damage does not need to be covered up at all. If an attack path requires damage to the part visible to the cardholder, the effort for hiding it—replacement parts (spare device), time needed, and skill to mask it—must be considered.

Small changes for repair of the device, such as added stickers or small increases in volume/size, do not make a device look materially different and are acceptable. Larger changes noticeably altering the shape of the device to a layman are not appropriate for consideration.

Attacks that require spare parts—e.g., plastic housings—can be obtained from a mechanical sample of the same device. These spare parts can be included in the attack costing as "standard specific parts" and can be included in the exploitation phase of an attack costing. A mechanical sample should not be included in the exploitation phase for access to the device and thus double counted.

Tamper evidence is not considered a suitable protection in and of itself. Attacks that damage the casing must consider hiding the damage through use of stickers, stands, or other such methods. With the advent of inexpensive 3D printing and home laser printers, there is not considered to be a limit to the damage that may be repaired through simple means and costed as “standard” parts using “standard” equipment.

Where sensitive data is exposed inside the PED or EPP, but outside of the immediate area of the keypad contacts—i.e., outside the keypad area—or internally within the security processor, this data must be protected by a tamper-responsive envelope. Use of physical barriers alone, such as plastic walls or tamper evident protections, is not considered sufficient to meet the requirements to protect customer PIN data or cryptographic keys. A POI that does not implement tamper-detecting side walls for its secure area must be implemented in such a way that the sensitive signals are otherwise protected with methods that go beyond purely physical and tamper evidence; otherwise, it fails the evaluation. This must include the use of a dedicated security processor which has internal protections against accessing signals on the pins or bond-out wires, with no sensitive signals exposed outside of this security processor that are not otherwise protected by a tamper-detecting envelope.

PIN Bugs

If a device includes front-case switches with guard rings as the only keypad (front case) security mechanisms protecting against the insertion of a PIN bug, then a Proficient level of expertise should be used in the exploitation phase of the attack for Requirement A2. If Expert level is accounted for in the exploitation phase, strong justification, including full testing on a sufficient number of samples, must be provided in the assessment. In most cases, the device must include additional types of security mechanisms protecting the front case of the device.

In most cases, only a Layman, Skilled, or Proficient level of expertise should be used for the installation and testing of a PIN bug during exploitation. It is expected that, during the identification phase, an attacker would develop a script executable by a Layman, Skilled, or Proficient person during the exploitation phase of the attack. If an Expert level is used for this phase of the attack, strong justification must be provided in the assessment, such as a full description of the specialized nature of the bug to be installed.

In general, the Identification phase should include a full dry run for the installation and testing of a PIN bug, resulting in a complete script to be followed in the Exploitation phase. In rare instances, additional steps may be required in the Exploitation phase because of nuances—e.g., slight variations in tamper-switch connections—between devices.

Given the recent rise of cheap electronic systems that provide integrated communications and processing options, any bug that would be used for detecting binary data (such as key pressed / not pressed or capturing data on an ICC I/O signal) must be considered to have a size of 15 x 15 x 5 mm, with a minimum width of 10 mm when costed as a standard part, which includes wireless communication capabilities. Use of a flexible keypad membrane to capture keypad signals must also be considered as no more than a standard part.

Where interception of very high-speed signals—that require impedance or transmission path length matching—is required, a specialized part may be considered. The lab must provide justification of the reasons for this in the report, including captures of the signals to be intercepted showing a phy bandwidth per trace in excess of 200MHz.

For the Identification phase for PIN-bug attacks, additional time spent analyzing the device under attack can be used in lieu of Restricted or Sensitive information. Restricted or Sensitive information should only be used when the total attack-potential calculation using Restricted or Sensitive information is less than the total attack-potential calculation using the additional attack time, such as through reverse-engineering.

Note on “parts”: *Although this document does not seek to outline all types of parts, it is considered that the vast majority of PIN bugs will be considered as standard parts, including those that use a flexible circuit mat to cover the keypad (as this is very common during PIN attacks).*

An Approach to Calculation

The above section identifies the factors to be considered. The table below gives guidelines for the individual factors.

For a given attack it might be necessary to make several passes through the table for different attack scenarios (for example, trading off expertise for time or equipment). The lowest value obtained for these passes should be retained. In the case of a vulnerability that has been identified and is in the public domain, the identifying values should be selected for an attacker to uncover that attack scenario in the public domain, rather than to initially identify it.

Table B2: Attack Potential Factors

Factor	Range	Identification Phase	Exploitation Phase
Attack time	≤ 1 hour	0	0
	≤ 2 hours	1	1
	≤ 4 hours	1.5	1.5
	≤ 6 hours	2	2
	≤ 8 hours	3	3
	≤ 12 hours	4	4
	≤ 16 hours	4.5	4.5
	≤ 24 hours	5	5
	≤ 40 hours	5.5	5.5
	≤ 60 hours	6	6
	≤ 100 hours	6.5	6.5
	≤ 160 hours	7	7
	Beyond 160 hours	7.5	7.5
Expertise	Layman	0	0
	Skilled	1	1
	Proficient	3	3
	Expert	4	4
Knowledge of the POI device	Public	0	0
	Restricted	2	2
	Sensitive	3	3
Access to the POI device per device required for the attack. ***Note:*** *If more than one device is required, the values must be multiplied by the factors given above.*	Mechanical sample	1	1
	Functional samples without working keys	2	2
	Functional sample with working keys and software	4	4
Equipment required for the attack	None	0	0
	Standard	1	1
	Specialized	3	3
	Bespoke	5	5
	Chip-level attacks	7	7

Factor	Range	Identification Phase	Exploitation Phase
Specific parts required	None	0	0
	Standard	1	1
	Specialized	3	3
	Bespoke	5	5

An approach such as this cannot take account of every circumstance or factor but should give a better indication of the attack potential. Other factors, such as the reliance on unlikely chance occurrences or the likelihood of detection before an attack can be completed, are not included in the basic model but can be used by a tester as justification for a rating other than those that the basic model might indicate.

Determining Applicable Time and Levels

For each phase, the testing laboratory shall document all necessary steps, including expertise, equipment, specific parts needed, and time required to operate (in hours).

This information is best summarized in a table containing all the items described above.

Attack Examples

In the following examples, attacks on a very well-designed device are described. These examples are intended to illustrate how attack ratings may be represented. New devices must be rated through direct testing and under no circumstances may have ratings relying on these examples. The device is very compact and there is limited space inside the device; therefore, it is assumed that a bug or its circuit has to be installed into a fake housing that provides more space than the original housing. Several examples of attacks are presented, including two for retrieval of PIN data entered on the keypad, one for retrieval of data from an ICC reader, two attacks on MSRs, and an SCA example.

Attack Example 1 – POI PIN-disclosing bug

For the first attack to insert a PIN-disclosing bug into a POI, the bug will be connected to the capacitive keypad controller. Here the backside security mesh is disabled (inner mesh attack, access to keypad controller). It is assumed that such an attack might be possible.

Steps for Identification:

1. The attacker must perform reverse-engineering of the device and develop the attack procedure. This step requires Expert knowledge of electronic engineering and the capability to perform the mechanical and electronic tests required. The device will cause an alarm during the development phase, since the case switches have to be opened. For the reverse-engineering of POI internals, only one mechanical sample is needed.

2. The attack has to be prepared by developing a PIN-disclosing bug for logging the clear-text PIN at the keypad controller serial I²C Bus. This bug must be very small to fit into the original housing or must be attached on the back case in a fake housing. The PIN-disclosing bug has to handle I²C Bus signals. The development of the dedicated bug hardware and software has to be performed by an Expert. Standard equipment and circuits have to be used to develop the PIN-disclosing bug for the keypad controller.

3. The housing has to be opened by removing a part of the backside. The corresponding keypad controller has to be accessed. Therefore, the security mesh must be circumvented without damaging or short-cutting this mesh. For the deactivation of the relevant parts of the security mesh,

a functional sample is necessary (the mesh is in most cases reparable in case of a shortcut or damage). The work can be performed by someone who is Proficient.

4. A PIN-disclosing bug is connected to the keypad controller serial I²C Bus signals of the POI through the PCB. To check whether the PIN-disclosing bug would work correctly, the I²C Bus signals are connected to the additional circuit with thin wires. The PIN-disclosing bug has to be attached on the back case in a fake housing, which holds the electronics and batteries for a bug. This requires professional skills with plastics (molding, forming, etc.) and standard equipment.

5. All indications of an attack have to be hidden with plastic repair measures. The manipulated device has to be tested.

Table used for Identification – Attack Example 1

Step	Expertise	Knowledge	Equipment	Parts	Samples	Time
1	Expert	Public	Standard	None	1 mechanical	12 hours
2	Expert	Public	Standard	Standard	Same mechanical	40 hours
3	Proficient	Public	Standard	Standard	1 functional with test keys	14 hours
4	Skilled	Public	Standard	Standard	Same mechanical	12 hours
5	Skilled	Public	Standard	Standard	Same functional with test keys	3 hours
Total	Expert	Public	Standard	Standard	1 mechanical 1 functional with test keys	81 hours

Steps for Exploitation

1. The housing has to be opened by removing a part of the backside, and the corresponding keypad controller has to be accessed. Therefore, the security mesh must be circumvented without damaging or short-cutting this mesh at two different locations. To accomplish this, several working steps with different success rates must be performed by someone who is Proficient:

   - Grinding through the GND and VCC plane;
   - Bridging relevant parts of the security mesh on the first mesh layer and cutting the bridged parts;
   - Grinding through the first already disabled mesh layer;
   - Bridging relevant parts of the security mesh on the second mesh layer (including the rerouting on layer 1) and cutting the bridged parts;
   - Grinding through the second, already disabled mesh layer to get access to the two I²C Bus lines.

2. A PIN-disclosing bug must be attached into the POI (functional sample with working keys): Therefore, the circuit has to be placed in the fake housing. The keypad controller has to be connected concerning the two I²C Bus lines with wires, silver-ink, etc. by someone who is Skilled.

3. Once the previous step is successfully achieved, the attacker will now attach the fake housing and hide all indications of the attachment with a plastic repair kit. The device has also to be tested. The device is now ready to be placed back at the target merchant location.

Table used for Exploitation – Attack Example 1

Step	Expertise	Knowledge	Equipment	Parts	Samples	Time
1	Proficient	Public	Standard	None	Functional with working keys and SW	8 hours
2	Skilled	Public	Standard	Standard (silver ink, glue etc.)	Same functional	1 hours
3	Skilled	Public	Standard	Standard (housing reparation)	Same functional	1 hours
Total	Proficient	Public	Standard	Standard	Functional with working keys and SW	10 hours

Combined Rating Table – Attack Example 1

Attack Potential for Inserting a PIN-Disclosing Bug for the keypad controller (mesh)

Aspect	**Identifying Value**		**Exploiting Value**	
Attack time	≤ 100 hours	6.5	≤ 12 hours	4
Expertise	Expert	4	Proficient	3
Knowledge of the device	Public	0	Public	0
Access to POI device	1 mechanical sample + 1 functional sample without target keys	3	Functional sample with working keys	4
Equipment	Standard	1	Standard	1
Specific parts	Standard	1	Standard	1
Attack potential per phase		15.5		13
Total Attack Potential		28.5		

Attack Example 2 – POI PIN-disclosing bug

This attack aims to insert a PIN-disclosing bug into a POI. The bug is placed at the IC card reader interface (DTR A13 "Penetration Protection"). It is assumed that such an attack is possible. The attack consists of the following steps:

Identification:

1. The attacker has to perform the reverse-engineering of the device to develop the attack procedure. This step requires Expert knowledge of electronic engineering and the capability to perform the mechanical and electronic test required. The attacker has first to identify the potential way to reach the I/O PIN of the IC card reader. Therefore, one sample of the POI will be opened. The location of the I/O PIN and the protection measures have to be analyzed. The device will cause an alarm during that development phase since the case switches have to be opened. For the reverse-engineering of POI internals, only a mechanical sample is needed.

2. The attack has to be prepared by developing a PIN-disclosing bug for logging the clear-text PIN at the I/O PIN of the IC card reader. The PIN bug must be very small to fit into a fake housing. The development of the dedicated bug hardware and software requires about 40 hours and has to be performed by an Expert with standard equipment.

3. The deactivation of the lateral tamper grid (flexible foil printed with conductive ink on two layers, integrated in a plastic shape filled with resin) may be performed by reaching the connector of the tamper grid to the security processor or by direct manipulation of the mesh. If the grid is deactivated, the attacker may drill a hole in the lateral side of the POI to insert a PIN-disclosing bug. The attacker will uncover the security grid lines of the flex foil in the lateral shape: The silver ink lines of the flex-foil connector will be connected to deactivate all security grids. Therefore, the plastic and the resin have to be removed from the flex foil. The attacker must drill a hole through the side of the POI through the opening in the lateral shape and contact the I/O of the ICC reader. For the deactivation of the flex-foil security grid, several functional samples are necessary (because the security grid is damaged and not repairable). The working space is highly limited, therefore additional space for—e.g., an endoscope is needed. This can be done by deactivating the lateral shape at both sides of the POI. The work can be performed by someone who is Proficient.

4. The space for a PIN-disclosing bug is highly limited inside the POI, therefore a fake housing must be produced to hold the electronics and batteries for a bug. This requires professional skills with plastic work (molding, forming etc.) and standard equipment.

5. A PIN-disclosing bug is placed into the fake housing. To check whether the PIN-disclosing bug would work correctly, the I/O PIN is connected (fixing/gluing) to the additional circuit with a thin wire with help of standard equipment (at least an endoscope with camera). The secure area is completely filled with electronics, complicating the attack. The manipulated device has to be tested.

Table used for Identification – Attack Example 2

Step	Expertise	Knowledge	Equipment	Parts	Samples	Time
1	Expert	Public	Standard	None	1 mechanical	12 hours
2	Expert	Public	Standard	Standard	Same mechanical	40 hours
3	Proficient	Public	Standard	Standard	2 functional with test keys	12 hours
4	Proficient	Public	Standard	Standard	Same mechanical	10 hours
5	Proficient	Public	Standard	Standard	Same mechanical	4 hours
Total	Expert	Public	Standard	Standard	1 mechanical 2 functional with test keys	78 hours

Exploitation:

1. The housing has to be opened from one side to get access to the lateral tamper grid without damaging the housing in a way that no tamper switches are activated. Additionally, the flex-foil tamper grid has to be disabled. Therefore, the covering resin has to be removed and three signals have to be bridged with six connections and disabled behind the bridge. As the working space is highly limited, additional space for, e.g., an endoscope is needed. This can be accomplished by deactivating the tamper grid at both sides of the POI. The work can be performed by someone who is Proficient.

2. A PIN-disclosing bug has to be attached into the POI (functional sample with working keys), and the ICC reader has to be connected to the additional circuit.

3. Once the previous step is successfully achieved, in this example the attacker will now attach the fake housing and hide all indications of the attachment with a plastic repair kit. The device has also to be tested. The device is now ready to be placed back at the target merchant location.

Table used for Exploitation – Attack Example 2

Step	Expertise	Knowledge	Equipment	Parts	Samples	Time
1	Proficient	Public	Standard	None	Functional with working keys and SW	5 hours
2	Skilled	Public	Standard	Standard (silver ink, glue etc.)	Same functional	1 hours
3	Skilled	Public	Standard	Standard (housing reparation)	Same functional	2 hours
Total	Proficient	Public	Standard	Standard	Functional with working keys and SW	8 hours

Combined Rating Table – Attack Example 2

Attack Potential for Inserting a PIN-Disclosing Bug for the I/O Pin of the ICC Reader

Aspect	Identifying Value		Exploiting Value	
Attack time	≤ 100 hours	6.5	≤ 8 hours	3
Expertise	Expert	4	Proficient	3
Knowledge of the device	Public	0	Public	0
Access to POI Device	1 mechanical sample + 2 functional samples without target keys	1 + 3	Functional sample with working keys	4
Equipment	Standard	1	Standard	1
Specific parts	Standard	1	Standard	1
Attack potential per phase		16.5		12
Total Attack Potential		28.5		

Attack Example 3 – Attack on the MSR

The following attack aims to insert an additional MSR head including a circuit to log the processed track data into a POI. It is assumed that such an attack might be possible. This attack consists of the following steps:

Identification:

1. The attacker has to perform reverse-engineering of the device to develop the attack procedure.
2. The attack has to be prepared by developing of a circuit for logging the track data at the MSR that fits into a fake housing, due to the low amount of space within the dongle. Alternatively, the attacker buys an appropriate bug on the Internet.
3. The attacker must develop a replacement circuit for the anti-skimming loop.
4. The device has to be opened for insertion of a second MSR head. This can be done by removing the magnetic card slide. Additionally, the anti-skimming loop measures have to be deactivated, and an extra frequency-simulating circuit has to be installed.
5. The bug has to be attached on the back case in a fake housing, which holds the electronics and batteries for a bug. This requires professional knowledge of plastics (molding, forming etc.) and standard equipment.
6. The track data is collected from the replaced MSR head and logged in the memory of the track data logging circuit—e.g., located in a fake housing—or sent to another device.

Table used for Identification – Attack Example 3

Step	Expertise	Knowledge	Equipment	Parts	Samples	Time
1	Skilled	Public	Standard	None	1 mechanical	2 hours
2a	Expert	Public	Standard (self-development of a bug)	Standard	Same mechanical sample	40 hours
2b	Skilled	Public	None (bug from internet)	Standard	Same mechanical sample	2 hours
3	Proficient	Public	Standard	Standard	1 functional with test keys	8 hours
4	Skilled	Public	Standard	Standard	Same functional with test keys	2 hours
5	Skilled	Public	Standard	Standard	Same mechanical sample	10 hours
6	Skilled	Public	Standard	None	Same functional with test keys	30 minutes
Total	a: Expert b: Proficient	Public	a: Standard b: Standard	Standard	1 mechanical 1 functional with test keys	a: 62.5 hours b: 24.5 hours

Exploitation:

In the exploitation phase, only three of the six steps are necessary:

1. The device has to be opened for insertion of a second MSR head. Additionally, the anti-skimming loop measures have to be deactivated and an extra frequency-simulating circuit has to be installed.
2. The bug has to be attached on the back case in a fake housing, which holds the electronics and batteries for a bug.
3. The track data is collected from the replaced MSR head and logged in the memory of the track data logging circuit—e.g., located in a fake housing—or sent to another device.

Table used for Exploitation – Attack Example 3

Step	Expertise	Knowledge	Equipment	Parts	Samples	Time
1	Skilled	Public	Standard	Standard	1 functional with working keys and SW	2 hours
2	Skilled	Public	Standard	Standard	Same functional	2 hours
3	Skilled	Public	None	None	None	30 minutes
Total	Skilled	Public	Standard	Standard	1 functional with working keys and SW	4.5 hours

Combined Attack Potential Rating for MSR Bug replacement – Attack Example 3

Aspect	Identifying Value		Exploiting Value	
Attack time	a: ≤ 100 hours b: ≤ 40 hours	6.5 5.5	≤ 6 hours	2
Expertise	a: Expert b: Proficient	4 3	Skilled	1
Knowledge of the device	Public	0	Public	0
Access to POI Device	1 mechanical sample + 1 functional sample w/o target keys	3	Functional sample with working keys	4
Equipment	a: Standard b: Standard	1 **1**	Standard	1
Specific parts	Standard	1	Standard	1
Attack potential per phase		a: 15.5 b: 13.5		9
Total Attack Potential		a: 24.5 b: 22.5		

Attack Example 4 – Attack on the MSR head

This second MSR attack aims to get access to the analog's pins of the coils inside the original MSR head. This attack consists of the following steps:

Identification:

1. The attacker has to perform the reverse-engineering of the device to develop the attack procedure.
2. The attacker buys an appropriate bug on the Internet.
3. The device has to be opened for manipulating the encrypting MSR head. Therefore, the magnetic card slide has to be removed. The attacker has to expose the analog pins of the MSR head.
4. The bug has to be attached on the back case in a fake housing, which holds the electronics and batteries for a bug. This requires professional knowledge of plastics (molding, forming etc.) and standard equipment.
5. The track data is collected from the manipulated MSR head and logged in the memory of the track data logging circuit—e.g., located in a fake housing—or sent to another device.

Table used for Identification – Attack Example 4

Step	Expertise	Knowledge	Equipment	Parts	Samples	Time
1	Skilled	Public	Standard	None	1 mechanical	2 hours
2	Skilled	Public	None	Standard	Same mechanical	2 hours
3	Skilled	Public	Standard	Standard	Same mechanical	0.75 hours
4	Skilled	Public	Standard	Standard	Same mechanical	10 hours
5	Skilled	Public	Standard	None	Same mechanical	.5 hours
Total	Skilled	Public	Standard	Standard	1 mechanical	15.25 hours

Exploitation:

In the exploitation phase only three of the five steps are necessary:

1. The device has to be opened for manipulating the encrypting MSR head. Therefore, the magnetic card slide has to be removed. The attacker has to expose the analog pins of the MSR head.
2. The bug has to be attached on the back case in a fake housing, which holds the electronics and batteries for a bug.
3. The track data is collected from the manipulated MSR head and logged in the memory of the track data logging circuit—e.g., located in a fake housing—or sent to another device.

Table used for Exploitation – Attack Example 4						
Step	**Expertise**	**Knowledge**	**Equipment**	**Parts**	**Samples**	**Time**
1	Skilled	Public	Standard	Standard	1 functional with working keys and SW	0.75 hours
2	Skilled	Public	Standard	Standard	1 functional with working keys and SW	1 hours
3	Skilled	Public	None	None	None	0.5 hours
Total	Skilled	Public	Standard	Standard	1 functional with working keys and SW	2.25 hours

Combined Attack Potential Rating for MSR manipulation – Attack Example 4				
Aspect	**Identifying Value**		**Exploiting Value**	
Attack time	≤ 16 hours	4.5	≤ 4 hours	1.5
Expertise	Skilled	1	Skilled	1
Knowledge of the device	Public	0	Public	0
Access to POI device	1 mechanical sample	1	Functional sample with working keys	4
Equipment	Standard	1	Standard	1
Specific parts	Standard	1	Standard	1
Attack potential per phase		8.5		8.5
Total Attack Potential		17		

Attack Example 5 – Keypad signals from bottom side

The next attack on the keypad signals is performed on another device with a less compact design.

Identification:

1. The attacker has to perform reverse-engineering of the device to develop the attack procedure. This step requires expert knowledge of electronic engineering and the capability to perform the mechanical and electronic tests required. The device will cause an alarm during that development phase since the case switches have to be opened. For the reverse-engineering of POI internals, which is calculated with 32 hours' working time—two days' reverse-engineering (expert) and two days' attack development (proficient)—only one mechanical sample is needed.

2. The attack has to be prepared by developing a PIN-disclosing bug for logging the entered PINs directly at the keypad matrix. The development of the dedicated bug hardware and software requires less than one week and has to be performed by an Expert with specialized equipment.

3. The two opening-detection switches at the bottom side of the device have to be circumvented to open the device and reach the bottom side of the Main PCB. It has to be ensured that the front casing is still pressed against the Main PCB to ensure that the keypad FPC and PCB are still pressed against the Main PCB. This can be done with standard equipment. For this example in the training, one additional mechanical sample is necessary. The work can be performed by a proficient attacker.

4. The two bottom mesh layers of the Main PCB have to be circumvented to get access to seven signal lines of the keypad matrix.

5. A PIN-disclosing bug is placed into the POI housing of a functional sample. In this example, the keypad matrix lines have to be connected (fixing/gluing) to the additional circuit with thin wires with help of standard equipment. The manipulated device has to be tested. The work has to be performed by a skilled attacker.

Table used for Identification – Attack Example 5

Step	Expertise	Knowledge	Equipment	Parts	Samples	Time
1	Expert	Public	Standard	None	1 Mechanical	32 hours
2	Expert	Public	Specialized	Standard (electronic parts, etc.)	1 functional with test keys	40 hours
3	Proficient	Public	Standard	Standard (silver ink)	1 additional mechanical	2 hours
4	Proficient	Public	Standard (optical microscope)	Standard (silver ink)	Same functional with test keys	4 hours
5	Skilled	Public	Standard (optical microscope)	None	Same functional with test keys	1 hours
Total	Expert	Public	Specialized	Standard	2 mechanical samples 1 functional with test keys	79 hours

Exploitation:

1. Disable the opening detection switches at the bottom side of the device. Removal of the bottom casing without triggering the switches at the top layer of the Main PCB.
2. Circumvent the two mesh layers to get access to the seven signal lines of the keypad matrix.
3. Attach a PIN-disclosure bug onto the keypad area and connect it to the key signal lines.
4. Once the previous step is successfully achieved, the attacker will now repair the housing that was damaged to circumvent the tamper switches. The bottom and top cases from the device in the identification phase can be reused. After testing the device, it is now ready to be placed back at the target merchant location. Standard equipment is required to implant the bug.

Table used for Exploitation – Attack Example 5

Step	Expertise	Knowledge	Equipment	Parts	Samples	Time
1	Proficient	Public	Standard	None	1 functional with working keys and SW	2 hours
2	Proficient	Public	Standard (optical microscope)	Standard (silver ink)	Same functional	3 hours
3	Skilled	Public	Standard (optical microscope)	Standard (silver-ink, wires etc.)	Same functional	1 hour
4	Skilled	Public	Standard (repair kit)	None	Same functional	2 hours
Total	Proficient	Public	Standard	Standard	1 functional with working keys and SW	8 hours

Combined Attack “Keypad Signals from bottom Side” – Attack Example 5

Aspect	Identifying Value		Exploiting Value	
Attack time	≤ 100 hours	6.5	≤ 8 hours	3
Expertise	Expert	4	Proficient	3
Knowledge of the device	Public	0	Public	0
Access costs	2 mechanical samples	3.5	Functional sample with working keys	4
Equipment	Specialized	3	Standard	1
Specific parts	Standard	1	Standard	1
Attack potential per phase		18		12 (fail)
Total Attack Potential		30		

Attack Example 6 – Side-channel attack

The last example describes a side-channel attack.

The attack aims at the determination of DES/AES and RSA keys used for various purposes—e.g., account-data encryption, key download, MACing, building the TLS session layer etc.—at the device using differential power analysis (DPA).

- A function of the device (API made available in the dedicated test firmware) is used to execute the DES/AES and RSA calculation with the key under attack;
- The data used for DPA is sent to the security processor using a temporarily available (only together with dedicated test firmware) serial test interface connector of the device.
- The power consumption can be measured with help of a current probe inserted between the voltage regulator output (VCORE voltage) and the through-hole-connection, which is used to connect a PCB inner layer for supplying the security processor with the VCORE voltage. The location that has to be manipulated is directly available on the security PCB (mainboard) without opening any housing of the device.

Identification:

The attack consists of the following steps:

1. Analyzing the security processor's electrical characteristics and pin connections (sensitive information is necessary because internal PCB design information and datasheet of the security processor is necessary). The attacker has to perform some reverse-engineering of the device to find the correct location for the necessary manipulation because of current probe insertion and to develop the attack procedure. This step requires knowledge of electronic engineering and the capability to perform the electronic test required (L3).
2. Determine the method to run DPA on the device. This consists mostly of analyzing the electrical and logical interface. This step requires expert knowledge of electronic and computer engineering.
3. Develop the attack set-up, including a trigger signal and the control to run the device in an automated way. The attack was facilitated building a test environment by executing DES/AES/RSA commands.
4. Perform the measurement.
5. Analyze the data samples in order to retrieve the key.

With about 650,000 measurement traces each (DES/AES) and 10,000 respectively 20,000 measurement traces for RSA, measuring the VCORE power consumption close to the security processor, it was not possible to disclose the key. With this result, the evaluators confirm the results of the security evaluation of the security processor. However, with the complete knowledge of the security processor internal countermeasures against DPA, it might be possible to find correlations with much more than 650,000 measurement traces for DES/AES or 10,000 respectively 20,000 for RSA, if the attacker invests more time than the evaluators. Therefore, the evaluators rated a successful attack within the following table:

Table used for Identification – Attack Example 6						
Step	**Expertise**	**Knowledge**	**Equipment**	**Parts**	**Samples**	**Time**
1	Proficient	Sensitive (schematic, layout); Restrictive (datasheet of SP)	Standard	None	1 functional with test keys	10 hours
2	Expert	Sensitive (API for external access and description)	Specialized (software development)	None	Same functional with test keys	20 hours
3	Expert	Sensitive (API for external access and description)	Specialized (software development)	None	Same functional with test keys	30 hours
4	Skilled	Public	Standard	Standard	Same functional with test keys	32 hours
5	Expert	Public	Specialized (software development for data analysis)	None	Not required	20 hours
Total	Expert	Sensitive	Specialized	Standard	1 functional with test keys	112 hours

Exploitation:

In this example, the exploitation phase, only three of the five steps are necessary:

1. Carefully insert the current probe between the voltage regulators output for the VCORE voltage and the via which is used by the VCORE voltage to enter the PCB.
2. Performing the measurement.
3. Analyzing the data samples in order to retrieve the key.

Table used for Exploitation – Attack Example 6						
Step	**Expertise**	**Knowledge**	**Equipment**	**Parts**	**Samples**	**Time**
1	Skilled	Public	Standard	Standard (glue, cables, solder. Measurement Probe)	1 functional with working keys and SW	1 hour
2	Skilled	Public	Specialized	None	Same functional with working keys and SW	32 hours
3	Expert	Public	Specialized	None	Not required	20 hours
Total	Expert	Public	Specialized	Standard	1 functional with working keys and SW	53 hours

Attack Potential for side-channel attack (DPA for DES/AES) – Attack Example 6				
Aspect	**Identifying Value**		**Exploiting Value**	
Attack time	≤ 160 hours	7	≤ 60 hours	6
Expertise	Expert	4	Expert	4
Knowledge of the device	Sensitive	3	Public	0
Access to POI device	Functional sample with test keys	2	Functional sample with working keys	4
Equipment	Specialized	3	Specialized	3
Specific parts	Standard	1	Standard	1
Attack potential per phase		20		18
Total Attack Potential	38			

Appendix C: Equipment Classification

Standard equipment

“Rule of thumb” followed: If it can be obtained easily (for example, over the Internet) for less than approximately US$1,000 or it can be reasonably expected that most people would have one or more (for example, a computer), it falls under this category. Most software is also to be considered standard, as it is highly likely that attackers will obtain illegal copies for free/little cost. Standard equipment would be expected for any attacks on the PCB or chip-carrier level, including exposure and attachment to chip bond-out wires. Unless otherwise indicated, the default classification for equipment is “Standard.” Furthermore, as equipment classified above Standard becomes more readily available over time, it will be appropriately reclassified.

- Cutting blades (for example, scalpels, X-Acto knives, box cutters, axes, saws, etc.), including both conductive and non-conductive blades
- Dremel tool, attachments, extension cable, holder/mill jig, etc.
- Screwdrivers of any type—including custom tips, as they can be cut from another type of screwdriver simply enough if not able to be purchased
- Hammers, saws, pliers, clamps (G-type, screw type, and any others)
- Oscilloscopes with two channels and external trigger inputs—up to around 1GHz with some leeway either way depending on buffers, bit-depth of A/D, additional features, etc.
- Simple milling machine
- Drills and drill press, any drill bits down to 0.1mm
- Conductive and non-conductive ink
- PCB etching equipment
- PCB design software
- Simple side-channel software (capable of performing time-displacement alignment, correlation analysis, basic time-frequency conversion and filtering, etc., and having a basic functioning user interface)
- Acid of most types used for attacks, including for etching of chips
- Solvents, cleaning materials, plastic working tools
- Low-resolution 3D printers
- Soldering irons of any type, including with heated rework tweezers or temperature-controlled hot-air guns for SMT (re)work
- Mechanical probes, dentist’s picks, tweezers (including non-magnetic), small mirrors, etc.
- Inspection microscopes up to 300x
- Multi-meters, continuity and resistance testers, IR and bi-metal temperature monitors
- Environmental chamber or equivalent heating/cooling tools (approximate range -80 C to 200 C)
- Current probes up to 1Ghz
- Simple EM probes (for example wire coils, antennas)
- Simple AM/FM receivers
- Custom jigs to hold devices/items during attacks (generally made from wood or 3D printed)

- Standard PC (either off the shelf or built from components)

PC-based instruments such as protocol sniffers, USB attached oscilloscope adapters, and graphical multimeters are considered standard equipment, especially if they do not require dedicated hardware or adapters.

Specialized equipment

“Rule of thumb” followed: If it can be obtained for between approximately US$1,000 and US$50,000, it falls under this category. Specialized equipment would be expected for any side-channel work on a modern CPU (for example, 32/64 bits, clock frequency >100Mhz) or hardware implementation. The values stated below are indicative only.

- Expensive, high-resolution, high-frequency, deep-buffer oscilloscopes (> 1GS/s, 1GHz, 16-bit, etc.)
- High-resolution 3D printers
- High-resolution milling machines (for example, for chip planing)
- Complex side-channel software, able to perform complex alignment beyond linear shift techniques, and relatively advanced, noise removal, etc., in accordance with the tool capabilities described in Appendix F. It is expected that this will be run on a standard PC (Standard equipment)
- Micro-probes for attachment to die-level features such as bus lines on chips
- Glitching systems, for example EM fault injection
- High-frequency/high-bandwidth EM probes
- 16/32-bit high-speed logic analyzer for capture and analysis of bus traffic
- Dedicated electronic cards
- Specialized test benches
- Protocol analyzers
- Microprobe workstation
- Chemical workbench
- Laser abrading/cutting

Bespoke equipment

“Rule of thumb” followed: If it cannot be obtained for less than US$50000 or extensive customization is required for such an attack, it falls under this category. Bespoke equipment is expected to be required only in very rare circumstances.

- Plasma etching equipment
- Automated (“push-button”) side-channel software to be used on a specific POI model to bypass strong countermeasures and leak keys from that specific model.
- Sophisticated X-ray 3-D imaging equipment

Chip-level equipment

“Rule of thumb” followed: Absolutely required for attacks at the feature level on chips (not bus level).

- Focused Ion Beam
- Electron tunneling microscope/Scanning Electron Microscope

Appendix D: Configuration and Use of the STS Tool

The NIST STS (Statistical Test Suite) is a reference implementation of the statistical tests described in *NIST SP 800-22 Revision 1a.*

The tester shall use NIST's STS tool, version 2.1.2 or later, or its mathematical equivalent. The tester shall verify that the compiled instance of the STS tool is operating correctly on the testing device by testing the NIST-provided sample data and comparing the results with those found in *NIST SP 800-22 Revision 1a (SP800-22r1a),* Appendix B. This configuration guidance is for use with STS version 2.1.2, though it will likely continue to be applicable to future versions.

A note on STS versions: *Prior versions of STS include bugs that have been fixed in the current version. Previous versions must not be used unless the critical fixes present in the current NIST tool have been backported. At a minimum, prior versions must disable the Lempel-Ziv compression test [Hamano 2009] and include fixes to the DFT (Spectral) test [Kim 2004], the Overlapping Template test [Hamano 2007], the Non-Overlapping test [NIST 2014], and the "Proportion of Sequences Passing a Test" test interpretation.*

The tester should request and obtain a sample of 2^{30} bits from the vendor. The tester should exercise care to verify that the vendor-supplied data is interpreted correctly by the STS tool (the STS tool assumes that binary data is in big-endian formatting on all devices).

The STS testing on the data shall be judged as a "pass" if it passes all of the tests, for both the "Proportion of Sequences Passing a Test" interpretation approach and "Uniform Distribution of P-Values" interpretation approach. If the data does not pass all tests, and the failure is marginal, the tester should acquire additional data from the vendor and repeat the testing, including both the initial data and the additional vendor-supplied data.

The STS tool should be configured as per guidance provided in SP 800-22 Revision 1a, which is summarized below.

Table C2: Configuration Settings

The following settings are consistent with the SP 800-22 Revision 1a document:

Configuration Item	Setting	Reference in Key Below
Length of bit streams (*n*)	1,000,000	[1]
Number of bit streams (sample size) *(M)*	1,073	[2]
Block Frequency block length	20,000	[3]
Non-Overlapping Templates template length	9	[4]
Overlapping Template template length	9	[5]
Universal block length (*L*), number of initialization steps (*Q*)	*L*=7, *Q*=1,280	[6]
Approximate Entropy block length	8	[7]
Serial block length	16	[8]
Linear Complexity block length	1,000	[9]

Key to Configuration Item Table Above

[1] *n* must be selected to be consistent with the requirements of all of the tests to be run. The Overlapping Templates, Linear Complexity, Random Excursions, and Random Excursions Variant tests all require *n* to be greater than or equal to 10^6 in order to produce meaningful results. The Discrete Fourier Transform (Spectral) test requires *n* to equal 10^6. (See SP 800-22r1a Sections 2.8.7, 2.10.7, 2.14.7, 2.15.7, and [NIST 2010].)

[2] The number of bit sequences (sample size) must be 1,000 or greater in order for the "Proportion of Sequences Passing a Test" result to be meaningful. (See SP 800-22r1a Section 4.2.1.) This value will be 1,073 for the first test, but any additional testing (for example, further testing to resolve test failures) will necessarily include more bit sequences.

[3] For the Block Frequency test, if n=10^6, the test block size should be set between 10^4 and 10^6. (See SP 800-22r1a Section 2.2.7.)

[4] The Non-Overlapping test requires selection of a template length of 9 or 10 in order to produce meaningful results. (See SP 800-22r1a Sections 2.7.7 and 2.8.7.) For a template length of 10, the MAXNUMOFTEMPLATES constant (in defs.h) should be set to at least 284 prior to compiling STS, otherwise most 10-bit aperiodic templates with a leading 1 bit are discarded.

[5] The Overlapping test requires selection of a template length of 9 or 10 in order to produce meaningful results. When n=10^6, the template size of 9 comes closest to fulfilling the parameter selection criteria. (See SP 800-22r1a Section 2.8.7.)

[6] The Universal test block length (*L*) and initialization steps (*Q*) must be consistent with the table in SP800-22r1a Section 2.9.7. For n=10^6, the only acceptable values are (*L*=6, *Q*=640) and (*L*=7, *Q*=1280). Note, any parameters passed into this test are discarded, and reasonable values are internally set. For n=10^6, STS automatically uses the parameters recommended here.

[7] For the Approximate Entropy (ApEn) test, SP 800-22r1a Section 2.12.7 requires the block length to be less than $\lfloor \log_2 n \rfloor - 5$. Other analysis [Hill 2004] has shown that for *n*=1,000,000, block lengths greater than 8 can cause failures more often than expected for large scale testing.

[8] The Serial Test block length is also set based on *n*. If n=10^6, the block length must be less than 17. (See SP 800-22r1a Section 2.11.7.)

[9] The Linear Complexity test block length is required to be set to between 500 and 5,000 (inclusive) and requires that $\frac{n}{M} \geq 200$. (See SP 800-22r1a Section 2.10.7.)

References

[Rukhin 2010] Rukhin, Andrew, et al., "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *NIST SP 800-22, Revision 1a.*

[Kim 2004] Kim, Song-Ju, et al., "Corrections of the NIST Statistical Test Suite for Randomness."

[Hill 2004] Hill, Joshua (InfoGard Labs), "ApEn Test Parameter Selection."

[Hamano 2007] Hamano, K. and Kaneko, T., "Correction of Overlapping Template Matching Test Included in NIST Randomness Test Suite," IEICE Trans. Fundamentals, vol. E90-A, no. 9, pp. 1788-1792, Sept. 2007.

[Hamano 2009] Hamano, Kenji, "Analysis and Application of the T-complexity." Ph.D. thesis, The University of Tokyo.

[NIST 2010] STS Software Revision History.
URL: http://csrc.nist.gov/groups/ST/toolkit/rng/revision_history_software.html.
Internet Archive:
http://web.archive.org/web/20150520193625/http://csrc.nist.gov/groups/ST/toolkit/rng/revision_history_software.html

[NIST 2014] Current STS Release Notes.
URL: http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html.
Internet Archive:
http://web.archive.org/web/20150103230340/http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html

Appendix E: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms

Approved Algorithms in connection with the requirements in this document are based on the approved algorithms listed in *NIST SP 800-57 Part 1 Rev. 4,* Section 4;

- Hash functions: Only algorithms from the SHA2 and SHA3 family are allowed, with output size >255.[1] MD5 and SHA-1 are not allowed for use.
- Symmetric-key algorithms used for encryption and decryption: AES must be used, with key size >= 128 bits or TDES with keys size >= 112 bits.
- Message authentication codes (MACs): CMAC or GMAC can be used with AES, as well as HMAC with an approved hash function and a key size >=128
- Signature algorithms: DSA, RSA (with PKCS1-v1.5 or PSS) and ECDSA with key sizes specified below.
- Approved key-establishment schemes are described in *NIST SP800-56A* (ECC/FCC[2]-based key agreement), *NIST SP800-56B* (IFC-based key agreement), and *NIST SP800-38F* (AES-based key encryption/wrapping).

The following are the minimum key sizes[3] and parameters for the algorithm(s) in question that must be used in connection with key transport, exchange, or establishment and for data protection in connection with these requirements. Other key sizes and algorithms may be supported for non-PCI payment-brand-relevant transactions:

	Algorithm				
	TDES	IFC (RSA)	ECC (ECDSA, EdDSA, ECDH, ECMQV)	FFC (DSA, DH, MQV)	AES
Minimum key size in number of bits:	112	2048	224	2048/224	128

[1] Except as noted, the use of SHA-1 is prohibited for all digital signatures used on the device in connection with meeting PCI security requirements. This includes certificates used by the device that are non-device specific and part of a vendor PKI, up to and including a vendor root certificate. The only exception to this is that the initial code on ROM that initiates upon the device start may authenticate itself using SHA-1, but all subsequent code must be authenticated using SHA-2.

SHA-2 or higher is recommended for other usages, but SHA-1 may be used in conjunction with the generation of HMAC values and surrogate PANs (with salt), for deriving keys using key derivation functions—i.e., KDFs—and random number generation. Where applicable, appropriate key length minimums as delineated in the Derived Test Requirements are also required.

[2] IFC: Integer Factorization Cryptography; ECC: Elliptic Curve Cryptography; FFC: Finite Field Cryptography.

[3] Other key sizes and algorithms specified in this appendix may be supported for non-PCI payment-brand-relevant transactions. They are also not applicable to the EMV kernel; cryptographic requirements for EMV Contactless transactions are set by EMVCo and/or the Payment Brands.

Key-encipherment keys shall be at least of equal or greater strength than any key that they are protecting. This applies to any key-encipherment keys used for the protection of secret or private keys that are stored or for keys used to encrypt any secret or private keys for loading or transport**. For purposes of this requirement, the following algorithms and keys sizes by row are considered equivalent**.

	Algorithm				
	TDES	IFC (RSA)	ECC (ECDSA, EdDSA, ECDH, ECMQV)	FFC (DSA, DH, MQV)	AES
Key size in number of bits:	112	1024	160	1024/160	–
	168	2048	224	2048/224	–
	–	3072	256	3072/256	128
	–	7680	384	7680/384	192
	–	15360	512	15360/512	256

TDES refers to TDES keys with non-parity bits. The RSA key size refers to the size of the modulus. The Elliptic Curve key size refers to the minimum order of the base point on the elliptic curve; this order should be slightly smaller than the field size. The DSA key sizes refer to the size of the modulus and the minimum size of a large subgroup.

TLS implementations must prevent the use of cipher suites that do not enforce the use of cryptographic ciphers, hash functions, and key lengths as defined in the Technical FAQs.

For implementations using FFC or ECC:

- **FFC implementations entities** must securely generate and distribute the system-wide parameters: generator *g,* prime number *p,* and parameter *q,* the large prime factor of (*p* – 1). Parameter *p* must be at least 2048 bits long, and parameter *q* must be at least 224 bits long. Each entity must generate a private key *x* and a public key *y* using the domain parameters (*p, q, g*).

- **ECC implementations entities** must securely generate and distribute the system-wide parameters. Entities may generate the elliptic curve domain parameters or use a recommended curve (see *NIST SP 800-186*). The elliptic curve specified by the domain parameters must be at least as secure as P-224. Each entity must generate a private key *d* and a public key *Q* using the specified elliptic curve domain parameters. (See *NIST SP 800-186* for methods of generating *d* and *Q*.)

- Each private key must be statistically unique, unpredictable, and created using an approved random number generator as described in this document.

- Entities must authenticate the FFC or ECC public keys using DSA, ECDSA, a certificate, or a MAC (see *ISO 16609 – Banking –Requirements* for message authentication using symmetric techniques. One of the following shall be used: MAC algorithm 1 using padding method 3, or MAC algorithm 5 using padding method 4).

IFC, FFC, and ECC are vulnerable to attacks from large-scale quantum computers. In 2017, NIST initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms, planned to end with a selection of new algorithms by 2023-2025.

Because of rapid progress in the field of quantum computing, it is advised to become informed/aware of this specific threat and its potential mitigations.

Appendix F: Side-Channel Analysis Standards for PCI-PTS Evaluations

***Note:** This appendix is for use in conjunction with DTR A7, "Non-Invasive Attacks for Cryptographic Keys."*

Objectives

Attackers' objectives are to compromise high-value cryptographic keys in a device by finding leakage. Testers' objectives are to search for paths to key leakage to be able to validate resistance to attacks. A test performed in accordance with the DTRs can result in no detectable leakage, may show how a key can be fully exposed but at an acceptable level of difficulty, or may produce a partial result. In every case, a test determining compliance will demonstrate that an attack in the field is infeasible with respect to PTS Security Requirements' pass/fail thresholds.

This appendix defines baseline expectations for PTS-recognized Laboratory competencies on side-channel analysis. Much of the information here is directly applicable to statistical testing on batches of device operations in the case of static key values. However other published analysis paths or example (but not restricted to) timing attacks, can be in scope. The PCI PTS standard assigns the highest rating level for attack calculations for the protection of PIN security-related keys; side-channel analysis is often a feasible method to compromise these assets. This appendix shall be used to provide consistency so that for all evaluations, all laboratories assess the effort of the side-channel component of an attack on any device similarly. The summary information below outlines expectations of recognized laboratories' competencies. That is: what laboratories must possess; what evaluations need to do; what reports should demonstrate. This information is also a reference for: laboratory recognition/maintenance, assessing the quality of evaluation test reports, and assessing laboratories' competencies.

Introduction

Side-channel analysis (SCA) is an important activity in PCI PTS evaluations, relevant to tests where the highest levels of confidence in security assurance are needed, for example, DTR A7. In these tests, it is necessary to derive an accurate measure of the effort to defeat a device using SCA techniques, distinctly from any physical penetration/modification.

Evaluations do not have to fully replicate SCA as would be applied in the field; a targeted and focused, "quick-scan" approach based on optimized practices capable of assessing key leakage is sufficient. The information below provides guidance and a framework for testing. In no circumstances should this be interpreted as a formula or checklist of necessary procedures.

PCI SSC expects all evaluation test reports to consistently provide good evidence of valid test results. Results must derive from robust testing and appropriate evaluation methodologies. The necessary foundation for a PCI PTS-recognized Laboratory's capability to deliver this includes the laboratory's core competencies in performing SCA: equipment, skills, experience, and capacity.

PCI SSC expects these competencies to be maintained and applied at an equivalent level to attackers capable of exploiting devices (including approved devices) using contemporary SCA-related methods and techniques.

This appendix provides an outline of SCA criteria to indicate PCI SSC's expectations regarding PTS-recognized Laboratory capabilities. This does not define a complete scope of the application of SCA and should not be interpreted as the boundary of what is adequate. This appendix provides supplementary information that can be referred to by evaluators, but it is not a tutorial. This memorandum may be used as a guideline for structuring tests and providing test evidence in reports.

The Laboratory General Requirements document includes relevant information on laboratory maintenance (competencies, audit processes, laboratory assessment of test artifacts, etc.). Other relevant documents include (but are not restricted to): DTRs, FAQs, Program Guide, Delta Evaluations Scoping and Guidance, and Report Templates.

Resources

All PCI PTS-recognized Laboratories must be capable of performing SCA with an attack potential that can distinguish compliant from non-compliant devices. This capability relies on a strong foundation of combined competencies that must be available for any test: skills, knowledge, and experience; collection and analysis equipment (both hardware equipment and software tools).

Laboratories' SCA resources must be maintained at levels consistent with industry "state-of-the-art" capabilities, including (but not restricted to): cryptanalysis of all relevant algorithms, higher-order DPA, and template attacks. Laboratories' resources must be improved regularly to reflect updates to the PCI PTS program, for example as version changes occur and/or as new attacks are published. Labs must have available a sufficient volume of personnel and equipment resources to be capable of performing SCA at any capacity of evaluations the lab undertakes.

The best SCA tools, even implementing advanced automation, are useless without a proficient user. PCI SSC expects laboratory personnel to generally participate in SCA-related activities such as (but not restricted to): training, teaching, research and development, contribution to new attack paths and techniques, and participation in relevant industry events. The following list outlines minimum expectations for evaluators.

Minimum expectations for evaluators

- Personnel performing SCA have a thorough understanding of known attack methods and techniques.
- Personnel performing SCA are sufficiently skilled and experienced to devise and implement new approaches.
- Personnel are experienced and capable of analyzing various types of devices, to perform SCA efficiently and optimally, adapted to different situations.
- Personnel performing SCA have a proven track record of being able to defeat devices, including successful removal of SCA countermeasures to extract unknown cryptographic keys.

PCI SSC expects laboratory hardware equipment and software tools to be fit for purpose for performing SCA. These resources must be sufficiently well developed to enable testers to determine devices' strengths and/or vulnerabilities, and to be able to produce evidence of this. The following list outlines minimum expectations for test resources.

Minimum expectations for test resources

- Configurable collection hardware and control software are capable of acquiring and storing high-resolution power end electromagnetic side-channel recordings.

- Triggering and noise-removal collection capabilities are able to overcome typical obstacles to acquiring good recordings, to achieve stable timing and good signal quality.
- SCA toolsets are demonstrably effective in analyzing, modifying, and extracting hidden information from side-channel recordings, including the capability to identify and remove SCA countermeasures.
- SCA toolsets are adaptable and extensible according to variable test situations, providing effective user interfaces and user-configurable functionalities.
- SCA toolsets are updated incrementally to improve performance, increase attack potential, remove bugs, and identify and resolve usability issues.

Laboratory capabilities related to any of the resources outlined above must correspond closely to information expressed in current versions of PTS documents, and to information releases from PCI SSC.

Principles and methods for testing

PCI SSC expects that the overall quality and effort required for SCA testing to conclude device compliance is comparable to the criteria described here.

SCA must not be hindered by lack of basic resources PCI SSC considers to be essential:

- Notwithstanding physical protections (such as those validated by DTR A1), approved devices must not contain cryptographic implementations which, under analysis, can be straightforwardly compromised through SCA.
- Devices provided for evaluations must be adequately prepared by the vendor for efficient lab testing.

SCA tests must not be hindered by the types of obstacles that experienced attackers having contemporary, available tools can overcome straightforwardly. The following non-exhaustive list gives examples of factors related to PCI SSC's expectations in testing.

Factors related to testing expectations – examples:

- Relevant information from the device vendor, such as: source code, functional documents, and other explanations of functionality, including accurate identification of hardware and software.
- Relevant assistance from the vendor, such as: devices adapted as necessary to support collection interfacing, signal acquisition access, collection triggers, etc.
- Availability of oscilloscopes, interface tools, workstations, RAM, data storage, etc.
- Signal-analysis capabilities such as GUIs that can display, manipulate, synchronize, superimpose, and compare multiple waveforms, including waveforms representing recordings and results derived from analysis of recordings. *
- Signal-processing capabilities to identify and overcome obstacles to SCA such as timing variation or other characteristics of recorded or derived waveforms, including deliberate countermeasures, or other noise that may obscure information. *
- Waveform-alignment capabilities that can accurately and efficiently synchronize specific sensitive events, or operations, occurring in algorithms under analysis.
- Waveform-leakage investigation capabilities, such as correlation, that can accurately and efficiently plot the dependencies of various data values processed in algorithms under analysis. *

- Configurable libraries for attacks on known or unknown cryptographic keys, implementing DPA and related cryptanalysis methods, and implementing robust analysis methods for: different leakage models, results interpretation, and graphing.*
- SCA toolset user interfaces and features such as: scripting, automation, process batching, tabulation of results, precise graphical rendering of recorded/filtered/results waveforms, information handling, and software-development capabilities, allowing evaluators to adapt analysis techniques and create new analysis methods. *
- Appropriate skills, experience, training, and available time for evaluators.
- Validation of any claimed SCA countermeasures as being both active and effective. Without these validations, tests cannot claim strong reliance on countermeasures.
- Validation of appropriate test methodologies sufficient to give a high degree of confidence that the device cannot be compromised using SCA methods similar to or different than those used in testing.

** Examples of specific attributes of SCA toolsets satisfying these expectations are listed below*

Critical steps in SCA scenarios

Even in straightforward scenarios—for example, in the absence of SCA countermeasures—several fundamental activities are necessary for SCA to succeed. Critical steps that should occur in most SCA scenarios are identified below.

- **Understanding of the algorithm(s) to be attacked and how they are implemented:** Determining the appropriate side channel(s) to acquire, collection method(s), and analysis tools and their configuration. Next, validating that the device behaves as expected before proceeding further; if necessary, gathering additional information and/or reconfiguring the test setup as required. This includes validation of hardware/software identifiers and claimed countermeasures.
- **Preparing the device:** While many of the considerations for the assessment of device physical penetration or modification are relevant to performing SCA, PCI SSC expects that in most cases the vendor's assistance must significantly ease the operation of the device and/or a specially prepared subset of the device (such as the security processor abstracted from the device). An important part of the evaluation is to determine which of these possibilities is optimum to assess leakage of the cryptographic implementation (hardware/software).
- **Acquiring stable (well-triggered) recordings at an optimum sampling rate for a particular analysis task:** The sampling rate should not be too low as to introduce aliasing, particularly if recording waveforms to input to attacks such as DPA/DEMA. Acquisitions should be set up to optimize SNR, amplitude, and time resolution. The evaluation should aim for well-timed, low-noise, high-resolution acquisitions appropriate to a particular task and commensurate to resources and techniques available to expert attackers.
- **Characterizing side-channel behavior through approaches such as SPA/SEMA:** A critical goal of this phase is to identify events in the algorithm under attack, to localize sensitive operations. Specific checks are necessary for certain test methods—for example, for EMA the relative location of the EM probe to the device's components is a highly sensitive variable factor. In many situations no useful features can be determined upon first inspection. In this case, it is necessary to explore either whether there are obstructions to SPA/SEMA that can be removed, and/or how the collection can be improved.
- **Alignment:** Time synchronization is a crucial step in almost all SCA attacks. For example, DPA attacks can often succeed straightforwardly after a small number of CPU clock cycles (or sample

points) have been well synchronized, but not without achieving this. Moreover, many aspects of timing variation in waveforms that outwardly appear to obstruct analysis—for example, randomly occurring delays—can in practice be removed straightforwardly. Hence, the evaluation should utilize the types of various alignment techniques available to expert attackers. A single application of alignment processing is often insufficient. In most situations, when a simple, or approximate, or limited-scope alignment test is performed, that approach is insufficient unless highly effective obstructions to alignment can be demonstrated.

- **Correlation:** Computations such as (but not restricted to) the Pearson product-moment coefficient are a crucial element in a test's capability to achieve alignment. Effective correlation capabilities are also crucial for identification of leakage that may often be obscured, for example to identify security-significant patterns in waveforms, test a device's relative susceptibility to leak internally processed data, localize sensitive operations for making well-targeted attacks, launch and develop attacks to infer secret data values. The evaluation should use correlation-analysis techniques equivalent to those available to expert attackers. If no correlation between a device's side-channel behavior and any non-sensitive/clear-text data values can be found, a possible explanation for this is that the test setup is not configured appropriately—for example, I/O data values unassociated directly with sensitive operations can be expected to leak unless there is a valid explanation otherwise. In most situations a test finding no leakage of any kind is insufficient unless valid reasons for this are identified.

  Establishing definite correlation of non-sensitive I/O data is important to (1) validate the collection setup and test methodology are not flawed, (2) to localize sensitive cryptographic operations, in developing attacks, and (3) demonstrate that cryptographic key data is more resistant to leakage than other data. The expectation is for most tests to be capable of establishing definite correlation of non-sensitive I/O data. Very strong justification for null correlation results is needed otherwise, in asserting device compliance.

- **Signal analysis and processing:** All side-channel recordings will contain a certain degree of noise in acquired waveforms—for example, ambient noise sources in the test apparatus and environment, including a device's physical architecture. Any noise source can be expected to introduce artifacts into side channels that obstruct an attacker seeking to identify exploitable signals such as: SPA/SEMA patterns, correlation leakage pinpointing sensitive operations, and key-dependent leakage. Many types of hardware-generated and algorithmic SCA countermeasures exist. When implemented appropriately, individual countermeasures that obscure sensitive signals can significantly delay or prevent SCA attacks. Combinations of well-implemented countermeasures therefore provide greater defense-in-depth. Nevertheless, unintentional leakage may become straightforwardly detectable when noise is removed. It is possible to utilize design information and source-code review, to some extent, to infer the likely effectiveness of obstructive noise. However, there is always a possibility that obstructions may be removed by analyzing and modifying recordings to reveal hidden signals. The degree of success for an attacker to achieve this can rarely be determined from simply viewing the appearance of initially acquired waveforms; obstacles may be trivial to overcome for an experienced attacker prepared to thoroughly investigate noise removal. Therefore, evaluations should usually include several activities such as signal filtering to explore and demonstrate the quality of claimed countermeasures, and to optimize testing.

- **Cryptanalysis:** Attacks such as DPA succeed by discriminating a few inferences of guessed key values from among many other possible values. One specific algorithm's key-leakage potential is unlikely to be similar to another algorithm or implementation. Detectable leakage is often highly dependent on the options an attacker pursues. Hence, a simplistic implementation of a published attack, alone, is often insufficient to validate a device's compliance; and the evaluation should make use of attack-modeling options available to expert attackers. Examples of these options include, but are not restricted to: selection functions (the calculation event in an algorithm under attack), statistical discriminants (the bit value(s) or Hamming weights, etc. being searched for—correlation functions—e.g., Pearson product-moment coefficient, differencing, and others) and options for ranking key guesses and modeling known key-value leakage. In most situations, when a small number of options for key searching are attempted, it cannot be assumed that attempting other options would not succeed, without strong justification.

- **Optimization:** There are many approaches that will improve the chance of SCA success, considering the application of critical steps such as those outlined above. Depending on the discoveries made early in analysis, it is likely that some of the steps performed should be adapted, branched, and repeated—for example, by iteratively re-applying well-chosen processes. Large data size may become problematic in some situations; however expert attackers are skilled in overcoming this unless robust obstacles are present—for example, analysis time may be significantly reduced by waveform compression filtering. To minimize elapsed time, in many situations analysis should be optimized by performing pilot tests, before launching SCA on larger data sets, through analysis of appropriately chosen sample point ranges, etc. Additionally, there are many computationally expensive analytical tasks that may be parallelized and run efficiently in the background (utilizing elapsed time) while appropriately focused analysis (directed effort) occurs.

- **Validation:** It is necessary for evaluations to check that critical steps performed in tests have the potential to succeed. It is often difficult to distinguish between a device's apparently robust resistance to attacks versus results deriving from a flawed or naive testing approach. For example, the absence of significant correlation leakage may be due to several situations that produce similar outcomes: errors in correlation calculations, bugs in analysis tools, inappropriate choice of analysis parameters, poor alignment, under-sampling or poor noise filtering, electronic noise introduced by the collection unintentionally, electronic noise deliberately generated by designed hardware countermeasures, obstructive noise from algorithmic signal-randomizing countermeasures, countermeasures for mathematical masking of specific sensitive data values, etc.. Hence, tests should include processes to differentiate between valid results or null results and demonstrate this.

Testing rationale and reporting

The outline information above defines many important aspects of testing. This should not be interpreted as an exhaustive list or as a definitive formula. PCI SSC expects SCA to be well targeted; that is, focused analysis considering the device's characteristics and making best use of adequate evaluation resources. It is not expected that all of these examples of techniques should be applied in any particular evaluation. More exactly, evaluations should demonstrably produce results deriving from well-balanced decisions corresponding to appropriate aspects of best practices.

In some situations, it is likely that testing from one evaluation can be reused straightforwardly in another evaluation. This situation occurs commonly when two devices having similar characteristics are evaluated by the same laboratory in parallel or in close succession. Other situations exist where SCA-test reuse, or partial reuse, is appropriate, including reuse of third-party testing. PCI SSC encourages optimized reuse of SCA by incrementally extending tests on similar devices evaluated sequentially. In these cases, enhancements and improvements in each new evaluation must be proportionate to these criteria and must be evident. In all circumstances where testing is reused, it is very important that the equivalent validity of reused testing to an entirely full-scope test can be demonstrated.

The outcome for an evaluation's SCA testing is to either defeat the device emulating a realistic attack scenario or derive a definitive rating showing that the effort for an attack in the field is clearly above the necessary threshold. If the latter is not achieved in a test, detailed evidence is necessary to demonstrate compliance otherwise. The evaluation test report is the principal resource for demonstrating compliance to Security Requirements. The report must be self-consistent and adequate to justify compliance on its own. Therefore, the report must provide good-quality evidence related to this appendix. This includes (but is not restricted to):

- Accurate identification of the device's cryptographic implementation (hardware and/or software) and accurate identification of any reused testing.
- Descriptions and justifications of valid test methodologies.
- Presentation of results including details of the testing performed and the device's behavior under test.
- Strong justification of conclusions showing compliance.

Overall scope of SCA testing

Generally, all the following secret and private keys, and related algorithms, are to be considered in scope of SCA testing:

- PIN encryption (PAN for SRED) for communication with both online host and external components,
- Remote key loading,
- Local key protection (internal and external memory encryption), and
- Authentication—e.g., firmware, display prompts
- Software loading

Determining test applicability

Valid statements of compliance cannot be made based on documentation/source code alone; applied testing is needed. Therefore, in principle, all algorithms used for protecting the assets listed above must be analyzed using SCA to some extent. At a minimum, this means recording algorithms' side-channel profiles and validating that the profiles agree with the algorithms' expected characteristics.

The evaluation should determine at least one algorithm to analyze thoroughly, based on a rigorous assessment of asset value versus feasible attacks. Reasoning for not testing any algorithm has to be explained. This reasoning should include reliable assumptions made in the vulnerability analysis based on asset value and attack complexity—for example, limits on collections such as delay insertions or key usage counters, and any additional countermeasures.

Prerequisites

To facilitate testing, the vendor has to provide "open" samples (with a wire or pin attached for triggering and, if applicable, attachments to locations agreed with the lab for power measurements). The vendor has to provide specially adapted firmware to allow cryptographic operations' input/outputs to be collected. For both EMA and power consumption (if possible), the measurement shall be performed directly at the chip/processor. The purpose of this is to identify how to optimize testing, including data gathering—e.g., probe placement—of a fully functional device. It is not valid to assume that physical barriers obstructing access to signals are in themselves sufficient to compensate for key leakage.

Effort to complete the overall device side-channel analysis

The descriptions below outline typical parameters that are acceptable for testing. Good attackers are creative and will not limit their efforts to below pre-defined limits such as disk storage size or an exact number of acquisitions, etc. PCI SCC expects the degree of any key leakage to be quantified using effective techniques—e.g., correlation, key attacks related to DPA, etc.—that are in agreement with these typical parameters but without reliance on parameter values as the primary justification for compliance verdicts.

For evaluating cryptographic implementations developed by the device vendor, the overall side-channel analysis effort should be expert-level work, assuming that well-prepared samples are available for analysis with effective tools, and assuming that testers are skilled in applying robust methodologies using these tools. Elapsed time is four weeks in most situations, for example to allow collections and computationally expensive processes to run automatically. EMA is generally expected to provide better results than other side channels; it is sufficient for a lab to rely on EM analysis only, if this decision is well-reasoned. Data collections size acquiring at least 100,000 high quality, high SNR traces is sufficient.

It is very likely that a significant part (even the majority) of the test effort will involve processing data—e.g., iteratively collecting, analyzing, filtering, aligning, performing correlation trials—following initial collections, to optimize inputs to key attacks.

Testing from separate evaluations meeting these best practices criteria can be reused, provided that (1) testing is no longer than from the previous major version of the standard—e.g., v5.x work can be reused if appropriate as part of a v6.x review—and (2) a complete chain of trust is demonstrated validating why previous testing is wholly applicable to a newly evaluated device. For "System on Chip" (SoC) type cryptographic implementations, where expert-level work, multiple effective SCA countermeasures, and collection sizes in excess of 1 million traces, etc., are demonstrated, test reuse will generally be straightforward to apply.

A good test may discover total, partial, or zero key leakage, feasible in a field attack scenario, and/or feasible in the white-box context of the evaluation. In situations where a feasible attack is known to be capable of exposing the key, the evaluation shall explain definitively the effort needed to extract the key and how compliance is assessed considering obstacles to key-leakage attacks. In assessing this, the evaluation must consider and justify how any degree of key leakage is considerably above an acceptable level, considering that successful attacks may require a lower degree of effort than the effort discovered through testing.

Reusing results

Tests identifying little or no correlation and/or algorithmic structure are generally insufficient for reuse. Since multiple devices commonly use same security processor or System on Chip for cryptographic operations, it can be useful to reuse results from other evaluations to decrease the efforts for side-channel analysis. Nevertheless, to ensure the transferability of results, the following points have to be considered:

- SCA results reused from other reports must not contain less information than required by the guidance.
- The measurement setup of the reused analysis has to be described in a way that it can be justified that this setup also applies for the device under evaluation.
- The identification and configuration of the cryptographic component has to be compared to ensure that the results also apply to the device under evaluation—e.g., registers for enabling/disabling countermeasures, etc.
- If results from third parties are reused, these third parties have to be approved PCI-labs. It is mandatory that the reports—i.e., the SCA part of the reports—are provided to the evaluation lab wanting to reuse the results.
- It has to be justified that the results remain valid. For example, no new attacks or measurement techniques are known which could impact on compliance.

- It should never be assumed that the use of the same chip will yield the same results given various implementation options across devices—for example, usage differences in the software libraries, compilers, hardware-versus-software engines, power filtering, PCBs, etc.

Examples of appropriate SCA tool capabilities

- Acquisition and storage for waveforms and associated data values recorded through interfaces to the test device and oscilloscopes.
- Automation and control for x-y-z positioning and scanning and configuration where EMA is being investigated.
- Scripting capabilities for waveforms' acquisition, analysis, processing, cryptanalysis, including factors such as command/response I/O, storage, and data editing.
- Display and rendering of multiple waveforms (both acquired waveforms and derived waveforms and derived graphs of results) at high resolutions with functions for: scaling, zooming, overlapping, etc.
- Alignment functions capable of matching and displacing waveforms against a reference pattern, including configurable acceptance/rejection criteria.
- Alignment functions capable of adapting/modifying waveforms in situations where straightforward matching and displacement is ineffective.
- Alignment functions capable of generating, storing, and applying waveform displacements and using references for matching derived from separately filtered waveforms.
- Correlation functions capable of calculating and plotting leakage of internally processed data (single and multiple bits) values across time-series and spectral waveform sets.
- Correlation functions capable of calculating and plotting matches between selected sample-point ranges within a waveform and across time-series and spectral waveform sets.
- Multiple cryptanalysis libraries such as first order and higher order DPA time-series sets and spectra sets, applicable to various algorithms used by evaluated devices, including configurable parameters to model different selection functions and discriminants, and capable of producing, analyzing, and graphing results.
- Waveform analysis and filtering capable of producing and plotting frequency spectra derived from time-series waveforms.
- Functions capable of combining, removing, deriving, or otherwise operating on waveforms and spectra within one set and between different sets.
- Waveform and spectrum filters capable of selecting, counting, removing, summing, or otherwise operating on sample points, amplitude ranges, or data values against configurable thresholds.
- Waveform and spectrum filters capable of calculating and plotting statistical analysis of data set, such as variance, standard deviation, average, absolute or values, etc.
- Waveform and spectrum filters capable of producing and plotting frequency-filtered outputs such as: high-pass, low-pass, band-pass/reject, harmonics series pass/reject, etc.
- Compression functions capable of reducing waveform storage size with minimal reduction in SNR.
- Ability to apply processing operations to signals during waveform acquisition.

- Ability to apply operations on signals in batches/sequences, and to parallelize and otherwise automate time-consuming processes efficiently.
- Ability for the user to configure the functionalities described above, including the ability to configure through GUI(s).
- Ability for the user to extend the functionalities described above, including the ability to develop innovative software.
- Ability for the user to output information, including graphical information, of sufficient detail and quality to demonstrate analysis findings, for example in evaluation test reports.

Glossary of abbreviations related to side-channel analysis

DEMA	Differential electromagnetic analysis
DPA	Differential power analysis
EMA	Electromagnetic analysis
SCA	Side-channel analysis
SEMA	Simple electromagnetic analysis
SNR	Signal-to-noise ratio
SPA	Simple power analysis

Appendix G: Domain-Based Asset Flow Analysis

Introduction

Modern IT solutions, including PCI POI, have reached a level of complexity that sometimes renders seemingly obvious questions as incomprehensible. A central question focuses on which components should be protected against which attack potential. In current solutions this often is hardly obvious and sometimes even surprises the developers.

The general idea is to determine when a certain asset is available at a certain location. Any hardware component or software module dealing with the asset will be virtually marked with the domain that the asset belongs to. We call this imaginary process "**tagging**." If a component or module is already tagged, it will keep the domain with the higher attack potential associated. Once the entire asset flow analysis is performed, the appropriate domain will be assigned for each software module, hardware component, and even PCB track. This allows the tester to apply the appropriate DTRs for the specific domain.

It is therefore expected that the developer of the PTS device will perform this domain-based asset flow analysis and provide the results and a proper explanation to the testers. The test lab will verify the analysis and use the effective domain rating as input for the evaluation.

For the vulnerability analysis—e.g., performed during attack costing according to Appendix B—it is furthermore vital to understand whether and when which asset may be attacked in a certain location. Obviously, the asset flow analysis also yields this information.

Assets and Domains

"Domains" is used herein as a shortened term for "security domains group assets" with similar protection requirements. Such domains are easily reflected by similar attack cost thresholds. While in general domains with the same requirements concerning attack potential might have to be kept segregated to reflect different administrative requirements, such a structure is as yet not foreseen in PCI POI. Therefore, the domain hierarchy is linear—i.e., any domain can always be represented by another domain with higher attack resistance. In effect, a developer can easily declare the entire hardware and software to belong in the PIN Key domain. This will save the developer from performing an analysis; on the other hand, some modules or components are likely to fail the protection requirements imposed by the PIN Key domain.

The PCI POI criteria defines various assets with corresponding protection requirements. Assets are defined by data and which kind of access it shall be protected against. Integrity means that any modification to the asset, whether spurious or induced, shall prohibit it being processed any longer. Confidentiality means that clear text of the data must not be disclosed. In Table H1 below, five Security Domains are defined from higher-protection requirements to lesser requirements:

Table G1: Security Domains Defined

Protection Level	Domain	Description
Higher protection ↕ Lower Protection	**PIN Keys**	Secret and private PIN-related cryptographic keys are protected to resist an attack potential of 35 points. All these keys must be protected to maintain integrity and authenticity and confidentiality. This pertains to both storage and usage of these keys. All keys used to secure information in the PIN domain, or other keys in the PIN Key domain, are included in this domain.
	Non-PIN Keys	All public keys, and secret/private keys used for securing operations in the data domain, must be protected to resist an attack potential of 26 points. All these keys must be protected for integrity and authenticity. Secret and private keys must also have their confidentiality protected as well.
	PIN	The cardholder PIN is protected to resist an attack potential of 26 points in DTR A1. PINs must not be disclosed. Passwords meet the same category. Although DTR A13 puts offline PIN at 20 points, the PIN shall consistently be considered as an asset class.
	Data	This class covers all other transaction data—e.g., PAN—and display prompts. The DTRs protect these assets at 20 points or less. The EMV L2 kernel sits here. PANs must not be disclosed, except as allowed in PCI DSS—e.g., ensure that the solution doesn't exceed the allowable brand truncation/masking guidance when displaying the IIN/BIN and trailing digits of a PAN. Both PAN and prompts must maintain integrity.
	Vendor	Any other data or functionality that is not related to the PCI POI DTRs. Compromising these assets cannot compromise PCI POI security. The corresponding security policy is largely at the discretion of the vendor.

It is vital that any lower domain cannot access any higher domain other than by dedicated, confined, and well-specified interfaces provided by the higher domain or any of its parents. Explicitly, the vendor domain has no access beyond itself, meaning that it is infeasible for any change of the code of this domain to gain any access to assets in a higher domain or change the behavior of any higher domain (except through the defined interfaces provided by those higher domains). It is not sufficient that the access is not foreseen, not implemented, or somehow prohibited by the code of the domain itself. If, for example, an application of the vendor domain requires user input, it must interface with the Data domain. The latter will enforce that this access cannot compromise the security requirements of PCI PTS POI.

If confidential assets are suitably encrypted using keys of at least the next higher domain, the encrypted asset is not a confidential asset. For example, keys from the PIN Key domain may encrypt other keys of the PIN Key domain, or of the Non-PIN Key domain. "Suitably encrypted" here means that the encrypting keys must at least have the same cryptographic strength as the keys they encrypt, use appropriate modes of operation, and meet any other PTS requirements such as the implementation of key blocks.

Note that the lack of suitable encryption does not necessarily imply non-compliance but does require that the encrypted asset remain in the same domain as if it is in clear text. For example, a battery-backed key may be used to encrypt all other keys for storage to provide tamper response—e.g. when the battery backed-key is erased on a tamper event all other keys are rendered unusable. However, if these encrypted keys are not stored in compliant key block formats, they would remain in their respective domains (ether PIN Key or Non-PIN Key domains).

Assets that require integrity or integrity-plus-authenticity protection may be accompanied by integrity/authenticity tokens—e.g., hash, MAC, HMAC, or a signature. Keyed tokens must be used for any asset that requires integrity plus authenticity. The critical time for an attack is the time span from the token validation to the processing of the asset. Additionally, unbundling must be prevented—i.e., it must not be possible to use a foreign token with the asset.

For example, it is valid to store keys in a domain of lesser security—if the keys are stored in compliant key-block formats and algorithms used for encryption and MAC are adequately strong. The modules and components performing the cryptography and processing the validated or clear-text assets necessarily belong to the PIN Key domain. This represents the idea of firmware in previous PCI PTS POI requirements.

Asset data that is properly protected by encryption and/or validation token, will be called an "**asset container"** in the remainder of this appendix.

Firmware and Process Spaces

While hardware tagging is quite simple—i.e., any single component may or may not process or store an asset—the situation is more complex with software. In this section we define some terms to help clarify the process.

Any code that could potentially endanger any asset—e.g., in case of malfunction or even complete replacement—**is considered "firmware" in the sense of the DTR**. At all times, firmware must be protected to a level of 26 points as required by DTR A4.

We define "**code"** as any instructions for a processing hardware, including microcode or netlists for programmable hardware, and any kind of data that may affect the processing by these instructions. Such data can be as simple as configuration bits, whether or not a certain security function is enforced. In PCI POI such data often is interpreted code, from simple access control rules represented as data for an engine up to large amounts of code—e.g., Java byte code in Android-based systems. As soon as data has the potential to compromise any of the DTRs, this data is considered firmware—i.e., code.

We define a process space as a container for code. Within the same process space there is no mechanism to ensure that any deliberate code change in one place cannot affect processing in another place. There are three established methods to segregate process spaces:

1. Segregation of process spaces by hardware support is well known from operating systems. Standard concepts comprise separated processors or CPU with memory management and execution modes of graded privileges. The proper configuration of the segregation hardware may not be overly simple, but in general it should be feasible to verify its effectiveness.
2. Segregation of process by simulation leads to a virtual hardware support. Standard concepts are interpreted programs like Java. Since such solutions often allow calls to native libraries—i.e., which are not interpreted, and the interpreter itself may be quite complex, in general it is much harder to verify that such segregation is actually effective.

3. Mechanisms must exist such that any memory released from a process space and reused or re-acquired by another does not contain any asset data.

Obviously, this implies that any loaders or managers for process spaces are effectively executed in the same process space as their children. Eventually any initial boot loader is the parent of all process spaces on the same hardware.

Hardware Tagging

All of the assets discussed above enter the PCI POI at some time using external interfaces—i.e., the devices are not produced with the assets built-in already.

- Consider the device in operational state and model each transaction type.
- Identify all interfaces involved in conveying assets or asset containers.
- Follow the path of asset containers until these are decrypted or validated.
- Consider explicitly how assets are conveyed internally between components.
- Define logical groups of hardware, including components, tracks, and wires carrying an asset with the corresponding domain.
- Define for logical interfaces a set of electrical signals and the passive components attached.
- Present all logical interfaces in a block diagram.

Software Tagging

A proven method to identify the domain for any piece of code is the following asset flow analysis algorithm:

1. Consider the software in operational state and perform all transactions—e.g., key loading, online, offline encrypted PIN, offline clear-text PIN, etc. Follow the incoming data from each external interface until messages are sent on external interfaces. Consider where assets or parts of them pass through the firmware. Tag each module with the corresponding asset domain.

2. For each module tagged, identify modules in the same process space and tag them with the highest asset domain found for the current module. The result is the domain structure in operational state.

3. Consider how the tagged modules are loaded during boot and updated during operation. Since modification of the module's code might compromise the corresponding assets, consider the authentication token of that code as an asset and assign it the same domain as the module itself.

   Since authentication tokens like digital signatures are self-protective, the relevant code of the loader might reduce to the code verifying that the image loaded is authentic. This includes any keys required to perform verification. This code may be active in non-operational phases, too. Furthermore, identify all firmware that manages the separation of process spaces. Tag these modules with the highest asset domain of the code authenticated or managed.

4. Repeat from step 2 until no new modules come into scope and no tagging has been changed.

Whatever software remains untagged by this algorithm is in the vendor domain—i.e., not firmware in the sense of the DTR.

A description of this analysis should be in the software architecture delivered by the vendor. It should make clear on the implementation level where the assets are produced, transferred, stored, and destroyed, giving a focused design description. It should furthermore provide a rationale how the process spaces in Step 2 are segregated. It will describe the authentication data required for code management and naturally describe the boot stack of the system.

Tag Coherence Verification

Identify which hardware components execute the modules tagged during software tagging. Tag the hardware with the highest domain tagged to any module executed. During the analysis it may happen that some hardware components have a lower domain assigned by initial hardware tagging than by software tagging. Identify the relevant assets that showed up in software tagging but were invisible during hardware tagging. Then do hardware tagging for these assets.

If both hardware and software tagging have been performed correctly, the effective domain for each hardware component should be identical. As a result, the domain of each hardware component, down to PCB tracks and for each piece of code—and even what is considered code in the first place—should be clear to the developer and the testers.

A description of this analysis should be in the hardware architecture delivered by the vendor. It should make clear on implementation level where the assets are produced, transferred, stored, and destroyed, giving a focused design description.

The vendor shall provide a block diagram at domain level that clearly identifies how the domains interact with one another. The corresponding programming interfaces or, if applicable, hardware interfaces shall be uniquely identified and documented. The diagram shall clearly identify whether software is executed on the same hardware or on separate CPU, memory, etc.

Applications

An application in the sense of a PCI PTS device is code that is for any reason not considered firmware. The developer produces guidance on how such code shall be developed and installs procedures to verify compliance with this guidance as a precondition for signing the code. Application code must have its authenticity verified by the POI firmware to be installed.

PCI PTS allows applications in the data domain and the vendor domain. Applications therefore can never handle clear-text PIN or keys of the PIN Key domain (although applications may handle keys in the Non-PIN Key domain, for example in the form of EMV public keys).

Applications shall be segregated from all firmware, including firmware of the same domain, and other applications at least from different issuers. This requires applications to run in unique process spaces. The tester shall verify that the separation of process spaces is effective for application sandboxing.

Asset Tagging Guidance

In this section, an incomplete list of typical assets and their respective domain is presented. Assets not listed here should be assigned similarly.

Table G2: Asset Tagging Examples

Asset	Protection	Domain
PIN encryption key	Confidentiality / integrity	PIN Key domain
Key decryption key	Confidentiality / integrity	PIN Key domain
Code validation public key	Integrity / authenticity	Non-PIN Key domain
PIN	Confidentiality	PIN domain
PAN	Confidentiality / integrity	Data domain
Display prompt	Integrity	Data domain
EMV public card key	Integrity	Non-PIN Key domain
Magstripe data	Confidentiality / integrity	Data domain
Key for magstripe data —e.g., SRED	Confidentiality / integrity	Non-PIN Key domain
EMV Level 2 code	Integrity	Data domain
Key matrix signals	Confidentiality	PIN domain
Touch Position	Confidentiality	PIN domain

Special Cases and Examples

An **encrypting magnetic read head** transfers magstripe data as asset containers—i.e., the data output is not considered an asset in its encrypted form. It must, however, store one or more keys for encryption and integrity protection. These keys belong to the Non-PIN Key domain, since the information it is protecting belongs to the Data domain (not to the PIN or PIN Key domain).

If the design can exclude that the display shows anything that the cardholder could somehow mistake for a **PIN prompt**, then **key matrix signals** and **touch position data** do not have to be treated as confidential.

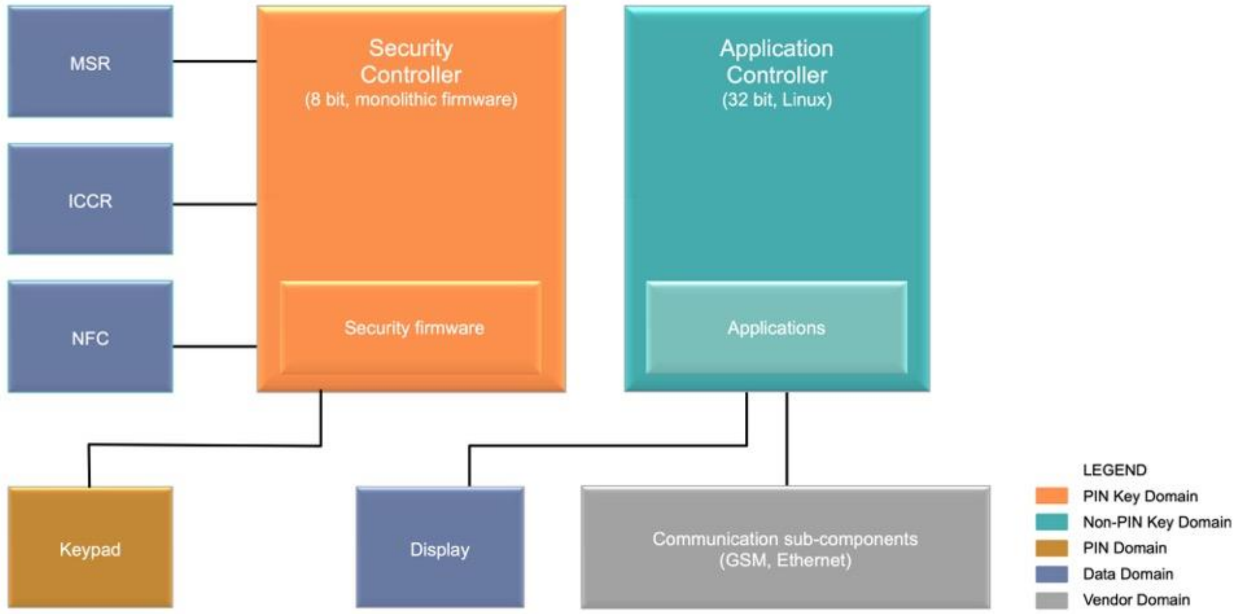
In common designs using a security processor (SP) and an application processor (AP), with the AP driving the display, the AP is at least part of the data domain. Even in cases where the SP is expected to verify the prompt before display, an AP will usually require the ability to authenticate its own firmware or application code. This will result in the AP being considered in at least the Non-PIN Key domain.

Example

Two-processor device example

This example assumes a common approach using a security processor to manage all cryptography, and an application processor to perform all complex interfacing—in this case with the unfortunate choice to attach the display to the application processor.

Figure G-1: 8-bit Microprocessor + 32-bit SoC with Linux


MSR
ICCR
NFC
Security Controller (8 bit, monolithic firmware)
Security firmware
Application Controller (32 bit, Linux)
Applications
Keypad
Display
Communication sub-components (GSM, Ethernet)
LEGEND
PIN Key Domain
Non-PIN Key Domain
PIN Domain
Data Domain
Vendor Domain

The 8-bit CPU runs a monolithic secure firmware. The 32-bit SoC runs a Linux OS managing all other software.

Table G3: Identification of Interfaces

Interface	Purpose	Handler (lowest level)
NFC Radio		Secure Firmware
Encrypting MSR	Read PAN for transactions from cards' magnetic stripes	Secure Firmware
ICCR	Read PAN for transactions from and convey PIN to ICC	Secure Firmware
Keypad	Enter PIN and Amount	Secure Firmware
Display	Display transaction information to cardholder and spam during idle time	Linux
GSM	Connect to payment host and terminal management	Linux
Ethernet	Connect to payment host and terminal management	Linux

Table G4: Identification of Operations

ID	Operation	Initiating Interface
1	Online PIN using ICC	Internal by amount entry
2	Online PIN using mag-stripe	Internal by amount entry
3	Online PIN using NFC	Internal by amount entry
4	Offline plaintext PIN via ICC	Internal by amount entry
5	Offline encrypted PIN via ICC	Internal by amount entry
6	Manual amount entry before PIN entry	Keypad
7	Amount from cash register before PIN entry	Ethernet or GSM
8	Remote key loading by payment system	Ethernet or GSM
9	Update of monolithic secure firmware by terminal management system	Ethernet or GSM
10	Update of peripheral microcodes by terminal management system	Ethernet or GSM
11	Update of Linux OS by terminal management system	Ethernet or GSM
12	Update of Data Domain code by terminal management system	Ethernet or GSM
13	Update of vendor domain applications by terminal management system	Ethernet or GSM

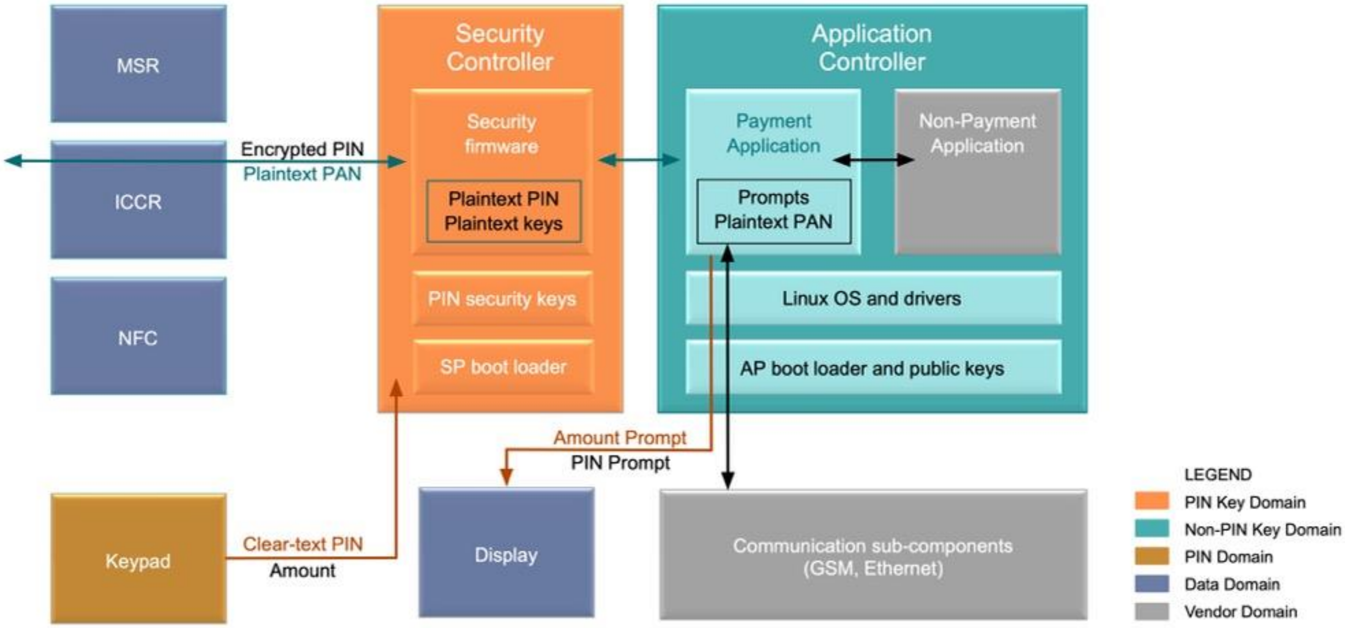
Offline PIN using NFC is not supported by the terminal.

Asset Flow Analysis Example

For the purpose of this example, we will not detail all operations and restrict ourselves to a representative subset.

Figure G-2: Asset Flow Description

Manual amount entry with offline encrypted PIN[1]



The merchant presses "START" on the keypad. The Security Controller firmware detects the key and sends a `START_TRANSACTION` message using the UART to the Application Processor. The latter is handled by an interrupt handler of the Linux OS, which maps it to the character device `/dev/ttyS1` for the process spaces maintained by Linux.

On the Application Processor `/dev/ttyS1` is acquired by the Payment Application. Upon receiving a `START_TRANSACTION` message, it returns a `QUERY_AMOUNT` message and displays an amount prompt via `/dev/pty1`, which interfaces to the display driver of the Linux OS.

The `QUERY_AMOUNT` message causes the Security Processor to scan the numeric keypad. Once amount entry is completed, the Security Processor returns `AMOUNT_DONE` to the Payment Application. The latter stores the amount and prompts for a card via `/dev/pty1`. It puts the Security Processor to card acceptance mode sending `WAIT_FOR_CARD`.

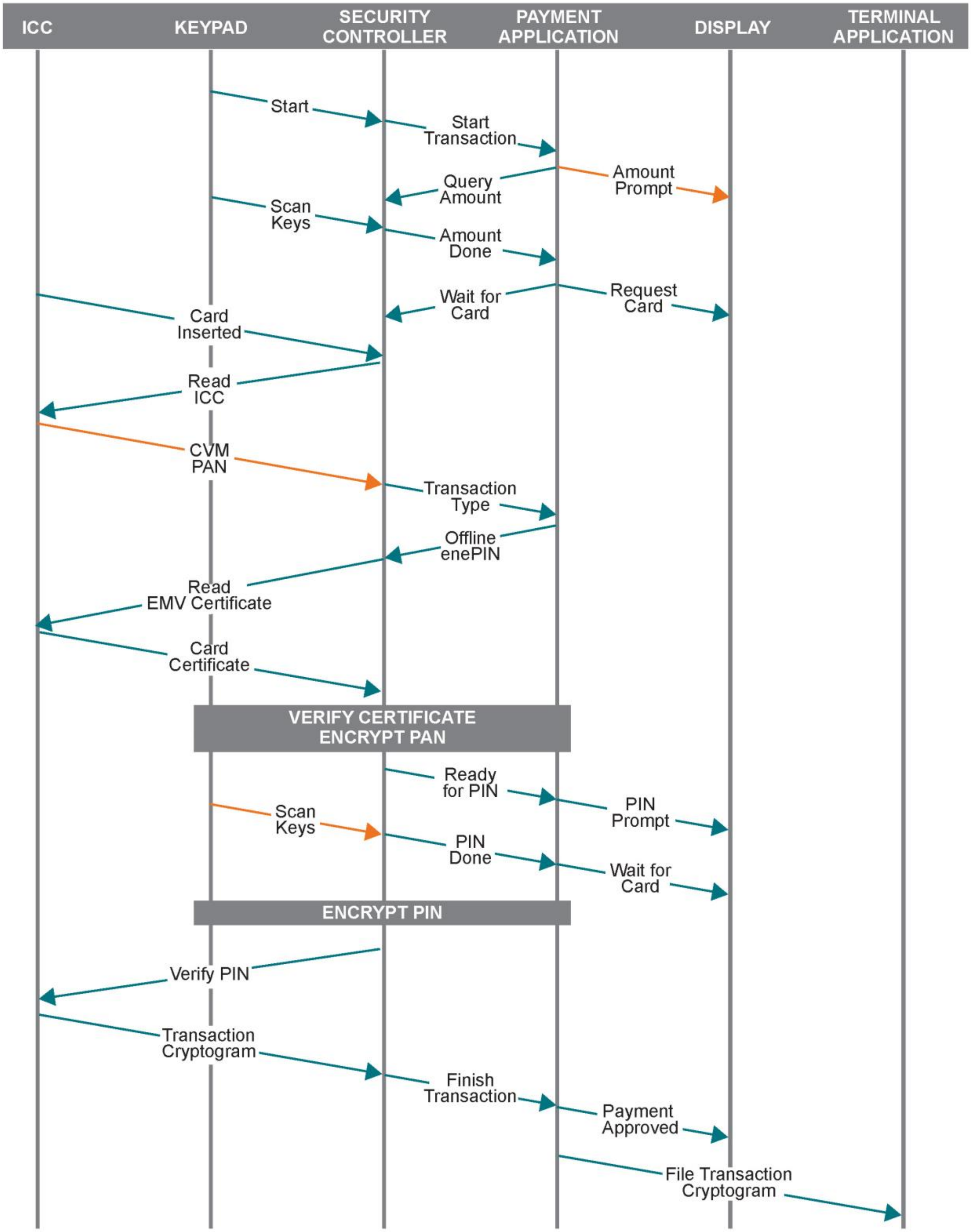
The cardholder inserts a card. The Security Processor detects this by polling the card switch. It selects contact payment and reads information from the card including the PAN, which is stored in the Security Processor. The latter sends CVM and Issuer Identification Number (IIN) to the Payment Application using `TRANSACTION_TYPE`.

The Payment Application selects the type of transaction and returns the decision using a `RUN_TRANSACTION` message – here with `Offline encPIN` as selected type.

[1] See Table G4, "Identification of Operations," lines 5 and 6, on previous page.

The Security Controller requests the ICC certificate and verifies it with respect to the database of issuer CA stored for the IIN in its internal database. If the certificate is valid, it encrypts the PAN for the IIN and sends `READY_FOR_PIN` to the Payment Application and activates scanning of the Keypad. The Payment Application displays a PIN prompt.

Figure G-3: Transaction Process Flow



When the cardholder finishes PIN entry, the Security Controller encrypts the PIN using the ICC public key. It sends `PIN_DONE` to the Payment Application, which displays a "Wait for card processing …" screen. The Security Processor has the card verify the PIN block. The transaction cryptogram, the encrypted PAN, and the verification status are returned to the Payment Application using the `FINISH_TRANSACTION` message.

The Payment Application displays the transaction outcome and creates a record for the payment processor containing Amount, Transaction Cryptogram, Time, Terminal ID, and encrypted PAN: This record is sent via the Unix domain socket `/var/run/terminal` to the Terminal Application for proper formatting. Unix domain sockets are handled by the Linux OS.

The terminal application will format the record according to the specifications of the particular processor and send the record to the payment processor using a network socket.

The network socket in turn is managed by the Linux OS. It wraps the message in appropriate TCP/IP frames and routes them via the proper interface. Depending on the connectivity, this may be the GSM modem or the Ethernet connection.

The GSM modem handles all GSM specific protocols internally. It is served by interrupt handlers of the Linux kernel, which handles it as a network interface. Similarly, the physical Ethernet layer is served by interrupt handlers of the Linux kernel and translated as a network interface.

Table G5: Asset Flow Analysis

The following table summarizes the asset allocation from the asset flow description:

Asset	Immediate	Asset Container	Domain
Prompt	Exists in Payment Application Conveyed by Linux OS to display		Data
PAN	Read from ICC Encrypted in Security Controller	Created in Security Controller Sent to payment processor via Payment Application, Terminal Application, Linux OS and GSM or Ethernet	Data
PAN Key	Resides in Security Controller		Non-PIN Key
Issuer CA Key	Resides in Security Controller		Non-PIN Key
ICC Public Key	Key determined valid after signature verification in Security Controller	Certificate read from ICC	Non-PIN Key
PIN	Read from Keypad Encrypted using ICC public key in Security Controller	Encrypted PIN block from Security Controller to ICC	PIN

Table G6: Asset Mapping

The following table maps these assets to components and shows tagging for hardware and software:

Component	Immediate Assets	Effective Domain
ICCR	PAN, encrypted PIN (vendor domain due to encryption)	Data
Security Controller	PAN, PAN Key, Issuer CA Key, ICC Public Key, PIN	PIN-Key
Keypad	PIN	PIN
Display	Prompt	Data
Payment Application	Prompt	Data
Linux OS	Application authentication keys, Payment Application, Prompt	Non-PIN Key
Application Controller	Linux authentication keys, Linux OS, Prompt	Non-PIN Key

All remaining components are tagged as vendor domain for this particular use-case. The effective tagging for each component is the most secure domain assigned.

Notes: *Security Controller denotes both the controller hardware and the monolithic software running on it. For simplicity it has been assumed that both Security and Application Controller use internal memories, only, or that the developer is willing to tag all external memory as the same domain as the controller itself.*

Appendix H: Evaluation Guidance for CPUs

Introduction

In PCI POI devices, CPUs generally process all sensitive information available inside the device. For this reason, chip vendors produced secure CPUs, which propose that all required security is built into the chip itself. This in general is misleading. Security CPUs may cover a subset of the relevant design issues, if integrated properly into the system. Moreover, for some CPUs this subset may be quite constrained.

The relevant security requirements as defined by the DTR are valid with respect to any CPU. Whether or not a CPU is properly used to cover or at least support any of these must be verified by the evaluators. This guidance shall be understood as a checklist, for which aspects should at least be considered. If a CPU is found to not cover any DTR entirely, the evaluators shall verify how other mechanisms are employed to bridge this gap.

The checklist applies to all processors inside any POI, which process any of the assets related to PCI POI Security Requirements—i.e., cryptographic keys relevant to PIN protection, clear-text PINs, clear-text PANs, display prompts, and clear-text card reader data. Please refer to Appendix G for definition of assets.

Processors that simply convey asset containers as defined in Appendix G can be left out of scope, provided that encryption or authentication is adequate according to the PCI requirements for the given asset. If any processor is excluded, the evaluator shall verify that this processor cannot compromise any of the DTR by whatever means.

Scoping

A processor is in scope if the Asset Flow Analysis as defined in Appendix G tags it to any non-vendor domain. As a minimum rule, a processor is in scope if at any time it:

- Processes clear-text keys, which includes using these keys, or passwords relevant to PIN or account-data protection;
- Processes clear-text PINs, PANs, or card-reader data;
- Verifies authenticity of key blocks or is otherwise able to mis-associate (encrypted) key values with key usage information;
- Verifies authenticity of prompts or is otherwise able to display prompts (or any other image) without further authentication;
- Processes clear-text input from keypads, touchscreens, or any other input device;
- Supervises environmental conditions relevant to the DTR testing;
- Monitors or conveys signals relevant to detect physical tamper or triggers tamper reaction;
- Monitors removal switches or triggers removal action;
- Can select key hierarchies—e.g., depending on tokenized pans.

Evaluators shall verify any rationale for taking processors out of scope in order to at least determine that none of these bullets applies.

Sensitive Signals

Sensitive signals are any signals tagged as any domain other than the vendor domain using the Asset Flow Analysis detailed in Appendix G. This applies to any clear-text keys, passwords, PINs, PANs or other cardholder data—e.g., from card readers—which are received at the contacts of the CPU. This also includes any key matrix or touchscreen position data. Attackers may try to eavesdrop these signals in order to compromise the associated asset. This even applies if these signals are somehow obfuscated or encrypted by keys that for any reason were excluded from testing. In this case the effort for de-obfuscation may add up to any point rating.

Furthermore, any tamper-detection signals, removal switch signals, prompts, or display signals are considered sensitive signals, as well as any PCB tracks signaling special modes that may be linked to sensitive services. Attackers may try to modify these signals to compromise the security of the device.

Evaluators shall examine whether any busses, in particular to external memory, may carry sensitive data (see also section about external memory below).

Most modern CPUs use a BGA-type packaging. The testers shall consider techniques to contact solder-balls. If a CPU is accessible, it can be feasible to contact any single ball. It may be harder if specific combinations are required, or if the CPU is hard to reach. The testers shall still consider whether any mechanical obstacles must be removed prior to attacking the BGA.

Even if a specific ball appears unreachable—e.g., some inner ball under a BGA using underfill—there may be other spots to reach the signal. Often the integrated carrier PCB of the IC routes the balls to the side of the package, where it may be easily accessible after some simple grinding. Even bond wires or bond-pads may offer an opportunity to contact otherwise well-hidden signals.

For other packages similar considerations may apply.

Tamper sensors

Security CPUs often support tamper or removal detection. Otherwise, these features are mostly implemented using GPIO and appropriate software. The corresponding signals available at the CPU contacts are sensitive signals in the sense described above.

For dedicated tamper sensors, the chip vendor may provide test results. The tester shall verify whether the sensors are

- Activated, since they may be disabled;
- Used in the tested mode and configuration;
- Active during all power management states of the device (including power-down detached from mains);
- Sufficient to fulfill the PCI POI criteria—e.g., a single signal static penetration-detection pin would fail PCI POI criteria.

Key scanning / Touch processing

Some CPU vendors claim that their approach to retrieve key presses or touch positions cannot be eavesdropped. This absolute position is never true unless input data are encrypted. However, there are various mechanisms, which by design are hard enough. Eventually the decision whether a scanning mechanism resists eavesdropping for 26 points can only be made after testing.

Similar claims are sometimes found for electromagnetic emissions of the scanning algorithm. Such emissions substantially depend on the track layout and potential peripheral electronics. They shall not be considered a property of the CPU.

External memory

Since code and data sizes tend to grow, some POI designs use external memory—i.e., storage that is located in an area with lesser tamper protection than the CPU. Obviously, no clear text of confidential assets must ever be stored in external memory that is reachable by an attacker. But access to memory can have various dangerous effects. Even modification of encrypted memory may be exploited for various goals. Analysis of address access may yield information about internal processing, which might for example be exploited for side-channel attacks. Clear-text code memory might be replaced by other chosen content, etc.

There are secure CPUs that support encrypted memory with authentication, which counter most of these attacks. A good address layout randomization and sufficient cache size may complicate address-bus analyses.

There are methods to store databases and unused code—i.e., files, authenticated and encrypted in memory—that may be accessible to attackers. Actual process memory is usually better protected physically.

If external memory is used, the testers shall describe in the report which kind of data is stored in this memory, at which time, and how it is protected. Note that processor native encryption solutions may not be adequate to, for example, store keys relevant to PIN or account-data protection. In any case, the verification of data authenticity must be performed within the CPU, and data integrity alone is not sufficient.

Security support functions

Environmental sensors

Most security CPUs feature sensors for supply voltages, clocks, and chip temperature as well as on-chip penetration sensors. The testers shall verify whether these sensors:

- Are activated, since they may be disabled;
- Cover all relevant power supplies and clocks of the CPU;
- Are active during all power management states of the device (including power down detached from mains);
- Are sufficient to monitor the environmental conditions of all sub-systems of the device.

Vendors tend to disable sensors due to reliability issues. Often the disabled sensors have to be substituted by some external sensor. The communication with this external sensor shall be regarded as a sensitive signal as defined above.

If any supply or clock is not monitored because the sensor is missing or disabled, the test lab shall conduct adequate testing for this signal.

Key Erasure

Most security CPUs erase some specific internal memory on tamper detection using hardware logic. Also common are inappropriate solutions, where erasure cannot be assured without power supply. In the scope of a PCI POI evaluation, it is usually hard to actually test the chip's internal erasure. Testers shall examine documentation of the chip to determine:

- When automatic erasure of memory is triggered;
- Whether it can be disabled;
- How erasure is performed;
- How long it takes until erasure is assured to be complete under any circumstances.

Random Number Generators

True random number generators are a common feature of security CPUs. Using some homebrew post-processing or even an entirely distinct generator is a common drawback implemented by POI developers. Testers shall verify that the generator is actually used, the data is not post-processed, and that the results have been validated by the relevant NIST tests. In most cases, re-running the test may be more efficient than referring to external test results.

Side Channels

There are CPUs with cryptographic libraries that were tested to not emit any side-channel information during specific cryptographic operations. These results are generally obtained in a dedicated measurement setup—i.e., not integrated in any payment device. If such CPUs are used, the tester shall verify the test report to determine whether:

- The relevant algorithms are supported and used properly—i.e., that the tested library is actually used and the corresponding user guidance is followed (this regularly requires source code analysis);
- All relevant emanation methods were tested in a sensible way—e.g., power analysis on the relevant supplies or electromagnetic emanations;
- The required depth of analysis was applied—i.e., at least all requirements from Appendix F, "Side-Channel Analysis Standards for PCI-PTS Evaluations," shall be met;
- The original testing is not older than three years.

If for any algorithm relevant to PCI POI any of these criteria fail, the tests for this algorithm shall not be re-used. The test lab shall re-test the algorithm instead.

If the CPU is encapsulated in some tamper envelope, testing may be restricted to data that can be gathered outside of the tamper perimeter; or penetrating the tamper protection may be counted as part of the attack.

Firmware and Secure Boot

Several CPUs contain ROM firmware. Mostly it is used for some kind of initial boot loader. Sometimes they even feature a BIOS-like library to more conveniently use SoC features. If used, and with the initial boot code there is no way around, this code is firmware. Software domain rating obviously reflects the highest asset rating processed by the CPU as determined by asset flow analysis defined in Appendix G.

The CPU vendor has to supply sufficient detail of the implementation to allow the tester to work through each DTR. This may include code samples if required by the DTR. Boot code as the trust anchor of the boot stack necessarily deals with keys. The exact key management is part of the PCI POI report and shall be available to the tester.

Test Re-Use

Security processors often claim various security features. Without exception, these features require the POI developer to correctly use and configure the CPU. This is the minimum effort for each test involving a security CPU and at least requires a clear and detailed guidance from the CPU vendor.

Beyond that, verification of the CPU vendor's claims in general is quite a time-consuming effort, which requires a lot of different hardware evaluation skills. It is therefore an expensive part of the testing, and it may be worth considering re-use.

In order to re-use test results, the original testing must be documented and be available to the test lab. The test description must clearly state:

- When and by whom testing has been performed;
- How the processor has been configured;
- How this configuration and choices for the test set-up may impact security, which may be represented as a dedicated user guidance for secure use of processor features;
- Whether and, if applicable, exactly which firmware—e.g., libraries—was used in the tests;
- What exactly has been tested;
- How test data was analyzed; and
- Why testing passed or failed.

If the detailed test procedures and results for whatever reason cannot be made available, PCI may accept reports from other PCI labs or attested to by trusted third parties—e.g., a CAST certification or a CC certification involving assurance to AVA_VAN.5. As an example, Table H-1 summarizes the information that shall be supplied to justify re-use.

Table H1: Test Re-use Justification Example

Item	Expected information
Original test date	29.02.2017
Test facility	XYZ (approved PTS lab)
Processor	ACME HighSec^{TM}
Configuration	HS-100U4-B The results are expected to be identical for all HS-100xx-x processors; the wildcards by position mean: 6 – is U if USB is supported, else A 7 – defines Flash size as 2^x GB 9 – defines packages: B: BGA, T: TSSOP All results about reaching contacts only apply to the BGA package, which is used in the current device as well.
Libraries	Crypto-Lib: ACHSCL.lib version 1.2.34.56 supplied by ACME Corp.
Scope	3DES side-channel resistance (DPA, EMA) AES side-channel resistance (DPA, EMA) High-temperature sensor Low-temperature sensor Voltage glitch resistance (V_{Core}, V_{Bat}) Clock glitch resistance (CPU & RTC) RNG Erasure of Keys on external tamper

The information in this table shall demonstrate that the quoted test results actually apply to the processor used in the current evaluation and that the functionality re-used is actually part of the scope originally tested. The final two bullets—how the test data was analyzed and why testing passed or failed—shall be examined by the test lab using the original evaluation report.

Appendix I: Security Policy Layout Example

Company Name

Model name

PCI PTS POI Security Policy

Date

(Security Policy Version Number if applicable)

Contents

Purpose

This should explain the purpose of the security policy, the PTS POI version to which the device is assessed, who should use it, and that any deviation from the approved use of the device will invalidate the PCI PTS POI approval.

General Description

Model Name and Appearance

- This should include the name of the model and where to locate the model name. This should also include a picture of the device and the label.

Product Type

- This section details how the device is to be used—countertop, hand-held, multi-lane, etc.
- This section states the POI Security Requirements version and Approval Class the device is evaluated and approved against.
- Other attributes of the device are listed here. Functions provided (MSR, ICCR, PIN entry, etc.) and the communication types (cellular, Bluetooth, Wi-Fi, etc.) on which the device is approved for use.
- For PIN entry devices that appear as a single device—i.e., where two physically and electronically distinct devices (e.g., a PED and a commercial off-the-shelf (COTS) device such as a mobile phone) appear as a single device through the use of the plastics to mask the connectivity—the guidance must state how the device meets the standard.
- For devices having beacons, the security policy states how those are compliant to the standard.

Identification

- This section defines and documents all hardware and firmware version options, both security and non-security relevant. Security-relevant options must be exhaustively defined and documented. For non-security options, the security policy must include a description of the option, but not a full list of all possible values.
- This section includes pictures of screen shots and procedures on how to locate the information both physically and logically on the device.

Installation and User Guidance

Initial Inspection

- This section provides user guidance to the merchant for physical inspection of the product to ensure the device has not been tampered or modified in transit.

Installation

- This section provides user guidance for how to install the equipment, installing cables, and any environmental considerations, such as security cameras or stands.

Environmental Conditions

- This section includes the environmental operating ranges including temperature, voltage, and humidity.

Communications and Security Protocols

- The guidance provides instructions for the proper use and configuration of any communication type and open protocol, including any security protocol.

Configuration Settings

- This section will provide all configuration settings necessary to meet the security requirements defined in this document.

Unattended Installation

- Guidance for unattended payment terminal designs where the ICCR slot cannot be positioned straight (horizontal) to the cardholder; the security policy stipulates the allowed installation height ensuring a sufficient view on the card-slot entry area

Handheld devices

- The guidance must state that if any handheld PIN entry device does not support SRED encryption, the system cannot be implemented to connect to a tablet or mobile phone, and any such use will violate the approval of the device.

Operation and Maintenance

Periodic Inspection

- Guidance and procedures are provided for continual visual and logical inspection of the product. An example would be to check the card slot for shim and look for wires, additional labels, or modification of the device plastics. This section should include picture and screen shots of the card slot.

Self-Test

- This section identifies all self-tests that the device performs, procedures that an operator may need to initiate, and the conditions under which each self-test is executed—for example, power up, periodic, conditional, on operator demand.

- Additional information must be provided for any device where the required memory re-initialization (security) cycle lasts longer than 24 hours. Specifically, describe how the firmware of the device during the cycle’s adjustment processes does not allow any security cycle to last longer than the combined maximum durations of the security cycle and the business cycle (48 hours).

Roles and Responsibilities

- This section identifies all roles supported by the device and indicates the services and permissions available for each role.

Passwords and Certificates

- This section contains specific details on how to change any default values, including passwords and certificates.

Tamper Response

- This section contains information on how the device will indicate a tamper event, and any requirements for the return of this device to the vendor for examination following such an event.
- This section includes pictures and screen shots as well as how the merchant is to react to a tamper event—i.e., stop using the device, call the help desk, notify your vendor etc.
- This section includes an illustration to show the user an actual tamper-response display message.

Privacy Shield

- If the device has been approved for use with a privacy shield, the guidance provides a picture of the approved privacy shield as properly installed and tested by the lab and how it is to be used.
- If the device has been approved without a privacy shield, the document must provide guidance on how to protect PIN entry.

Patching and Updating

- This section provides guidance on device update and patch procedures required for the secure operation of the device, including the ability to determine when the most recent patch was applied. The guidance also includes both local and remote update and patch downloading procedures for software, firmware, and configuration parameters.

Decommissioning

- This section provides procedures for the decommissioning of devices that are removed from service, including the removal of all keying material that could be used to decrypt any sensitive data processed by the device. The procedures may differentiate between temporary and permanent removal.

Security

Software Development Guidance

- This section contains references to any software-development guidance required for compliance, if applicable, with Open Protocols and SRED requirements. This documentation must clearly outline which functions, APIs, or modes of operation of cryptographic functions (such as cipher suites) have been evaluated by the PTS laboratory for securing cardholder data.

SSL

- If the device supports SSL, the security policy must clearly state that it is inherently weak and should be removed unless required on an interim basis to facilitate interoperability as part of a migration plan.

Signing

- This section provides guidance how any signing mechanisms must be implemented. This must include any “turnkey” systems required for compliance with B16, or any mechanisms used for authenticating application code as assessed under Requirements B4.1.

Account-data Protection

- This section details any account-data protection schemes employed—e.g., algorithms used, format-preserving encryption techniques—and whether the device supports the pass-through of clear-text account data using techniques such as whitelisting.

- The guidance must include procedures and use cases for devices that allow the enablement (turning on) or the disablement (turning off) of SRED functionality.

Algorithms Supported

- This section contains specific details on the cryptographic algorithms (TDES, SHA-2, etc.) and key-management methodologies supported by the device. It must detail the specific keys and usages of these keys for all key-management methods exposed to the device operators.

Note: *Key-management operations that are only used within the device, or between integrated device components, are not required to be detailed.*

Key Management

- The section clearly outlines the exact details of the key-management systems supported by the device—i.e., simply using the term "MK/SK" is not sufficient—and specifies that use of the device with different key-management systems will invalidate any PCI PTS POI approval. Include key name, purpose/usage, algorithm, key size, form factor in which key is loaded to the device, number of keys of each usage type supported, etc.

Key Loading

- This section contains specific details and procedures on how key loading must be performed for operation of the device. This must include any requirements for dual control and split knowledge.

Key Replacement

- The guidance states that keys should be replaced with new keys whenever the compromise of the original key is known or suspected, and whenever the time deemed feasible to determine the key by exhaustive attack elapses, as defined in *NIST SP 800-57-1.* The guidance should also include whom the merchant must contact and the contact method.

Acronyms

- List all acronyms used in the document.

References

- List all documents referenced such as user manuals, development guidance documents, and operation documents.