Payment Card Industry (PCI)

Software-based PIN Entry on COTS (SPoC)™

Test Requirements

Version 1.1

June 2020

Document Changes

Date	Version	Description
February 2018	1.0	Initial release
June 2020	1.1	Updated to align with publication of PTS POI v6.0.

Table of Contents

Introduction

Background

In traditional merchant payment PIN entry scenarios, cardholder authentication data—e.g., a PIN—is entered into a device specifically designed for the protection of PIN data—e.g., a PIN entry device (PED). The payment industry recognizes PEDs that have been independently tested and comply with detailed security requirements developed by PCI to ensure the confidentiality, integrity, and availability of the PIN data. Traditional PEDs rely on hardware protections as the primary mechanisms to ensure the security of PIN data within the device. Merchants use traditional PEDs to support cardholder PIN acceptance.

Mobile payment acceptance enables the processing of payment transactions – e.g., using a smartphone or tablet – that performs the functions of an electronic point-of-sale terminal. Software-based PIN entry solutions allow for the processing of mobile payment-acceptance transactions with a cardholder's PIN.

Software-based cardholder authentication provides an alternative for chip and PIN markets by providing a software application interface on a merchant's COTS device to capture and process a cardholder's PIN. These solutions rely on a combination of mechanisms and security controls including but not limited to device hardware, application software, and independent management and oversight of the entire process to ensure the security of the transaction and PIN data.

Please note: *This standard does not supersede other PCI standards, nor do these requirements constitute a recommendation from the Council or obligate merchants, service providers or financial institutions to purchase or deploy such solutions. As with all other PCI standards, any mandates, regulations or rules regarding these requirements are provided by the participating payment brands.*

The following figure highlights the security model differences (and reliance on controls) between a traditional PED implementation (hardware PIN) and a software-based PIN entry solution (software PIN) for the protection of PIN data.

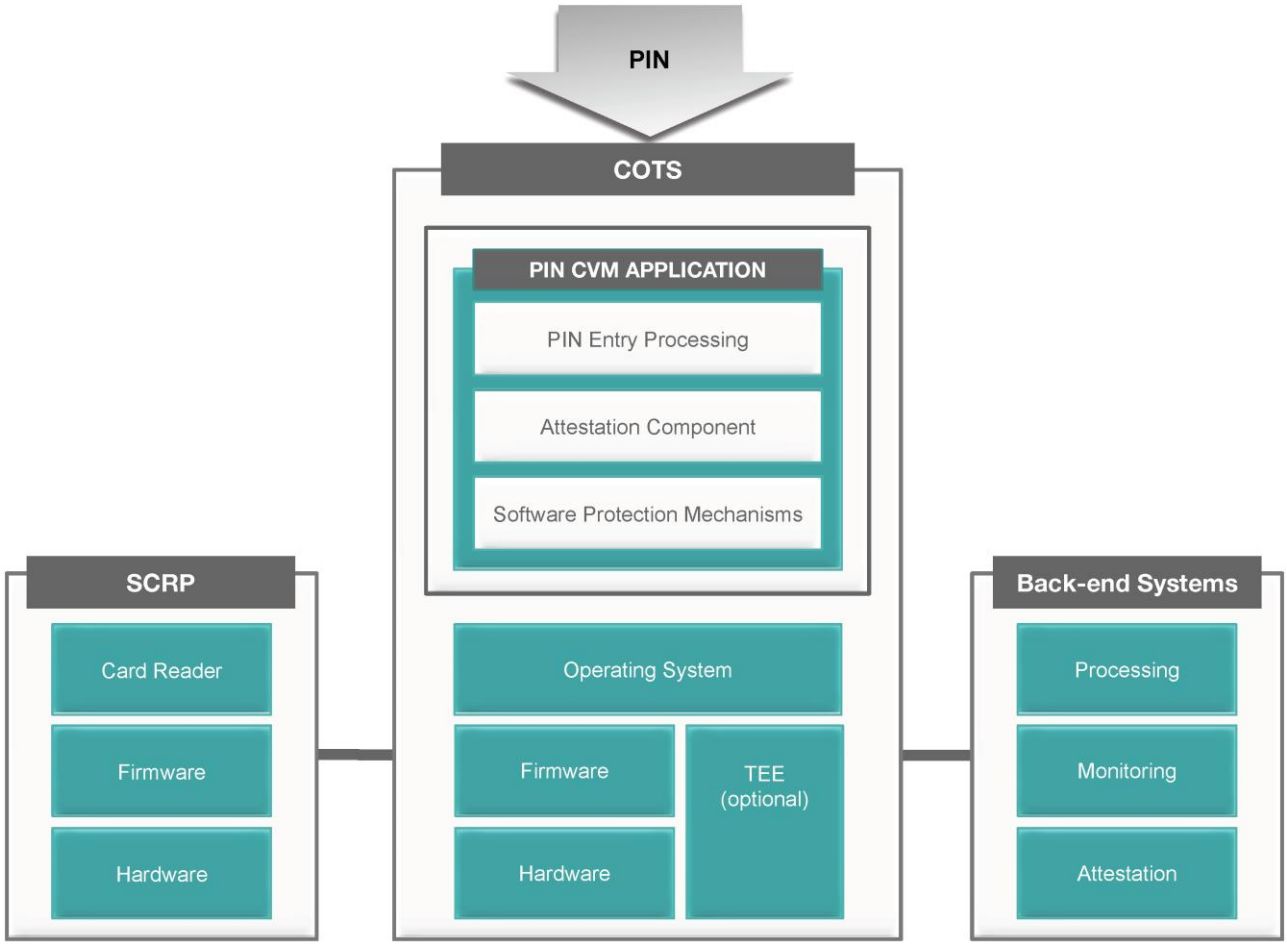


Figure 1: Example of PIN CVM Solution Architecture

Audience

This document is designed to provide more granularity and visibility into the testing processes performed by the evaluation laboratories that will perform the validation testing of the software-based PIN CVM solutions.

A separate document, *Software-based PIN Entry on COTS Security Requirements,* is available to define the security requirements for entities developing PIN CVM applications, organizations managing and deploying PIN CVM solutions, testers, and assessors. Solutions may optionally include magnetic-stripe readers that meet the security and testing requirements detailed in Payment Card Industry (PCI) Software-based PIN Entry on COTS Magnetic Stripe Readers Annex.

The testing requirements and security requirements have been developed for specific audiences, therefore it should be noted that there is not a one-to-one mapping between the documents. The two documents should be read in combination to fully understand the requirements and the methods for evaluation.

Software-based PIN Entry Solution Overview

The following diagram illustrates the flow of a PIN transaction in a software-based PIN entry solution. Steps 1-7 are detailed below.

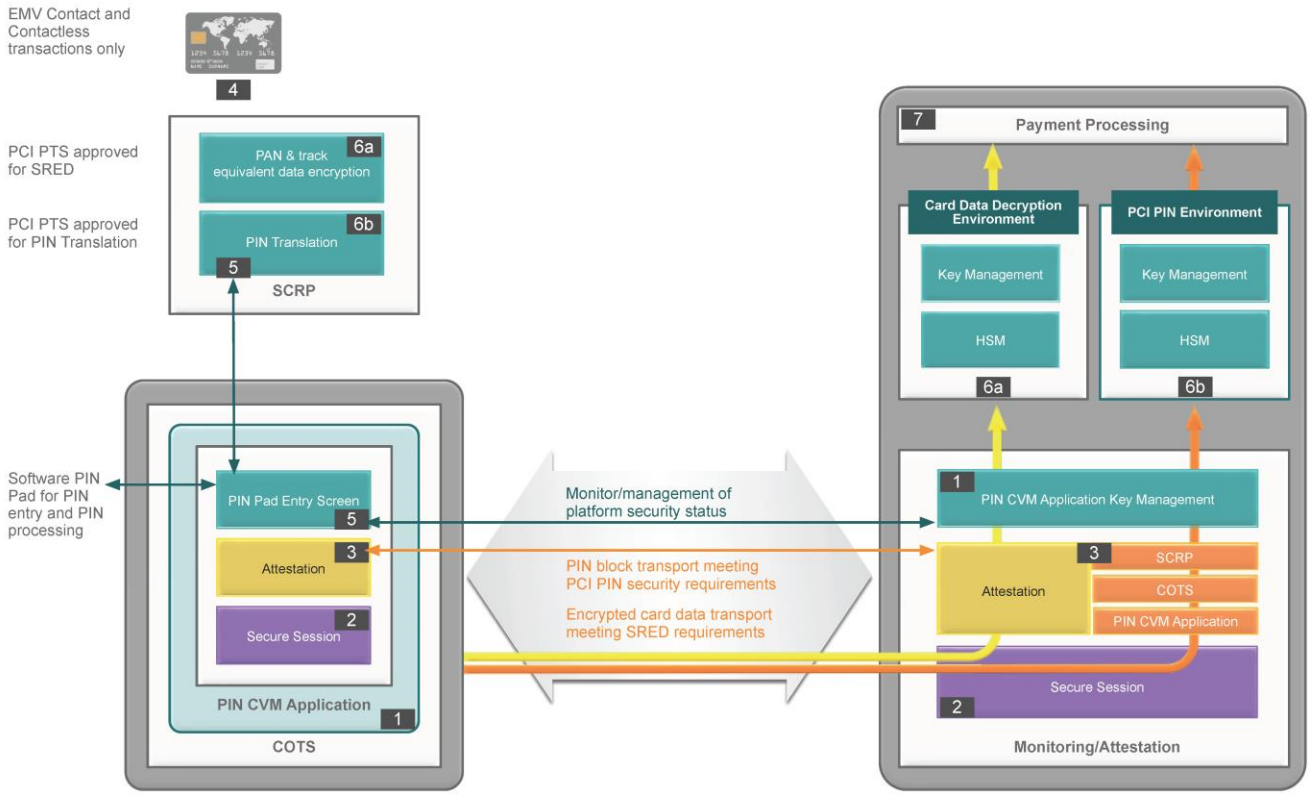


Figure 2: Software-based PIN Entry Solution

1. PIN CVM application and SCRP are initialized with their financial keys (this may be asynchronous with the transaction).
2. A secure communication channel between the PIN CVM application and the back-end monitoring system is established.
3. The back-end monitoring system determines the security status of the mobile payment-acceptance platform (SCRP, COTS platform, and PIN CVM application) using the attestation component.
4. An EMV card, contact or contactless or an NFC-enabled mobile EMV payment device is presented to the SCRP.
5. The PIN CVM application PIN entry component renders a PIN entry screen on the COTS platform and the cardholder enters their PIN using the rendered PIN pad from the PIN CVM application. The resulting information is enciphered and sent to the SCRP by the PIN CVM application.
6. The SRED component of the SCRP enciphers the account data using preloaded data-encryption keys according to either Figure 3 (online PIN verification) or Figure 4 (offline PIN verification) below.
7. The payment transaction is processed.

Figure 3:
Online PIN Verification

- An enciphered PIN block is generated by the SCRP component (online PIN processing, contact and contactless EMV) using preloaded PIN-encryption keys.
- The SRED data and enciphered PIN block (online PIN processing only) are transmitted to an HSM within the back-end payment processing system.

– OR –

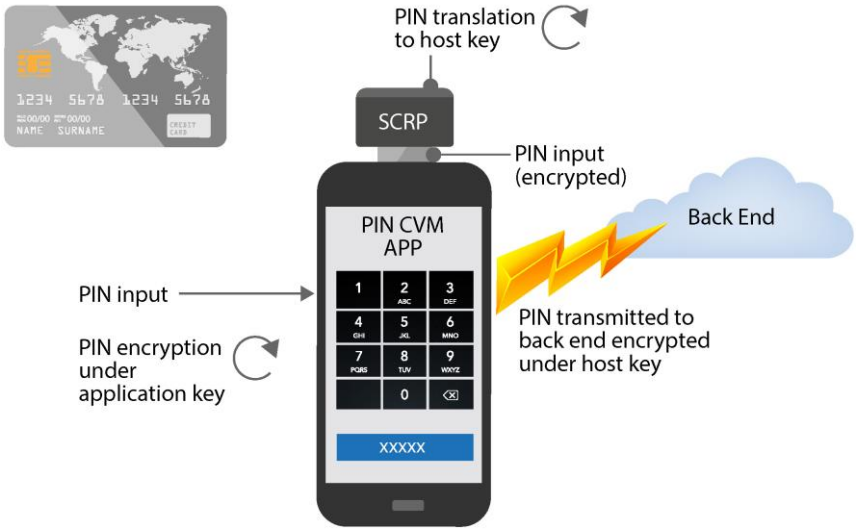
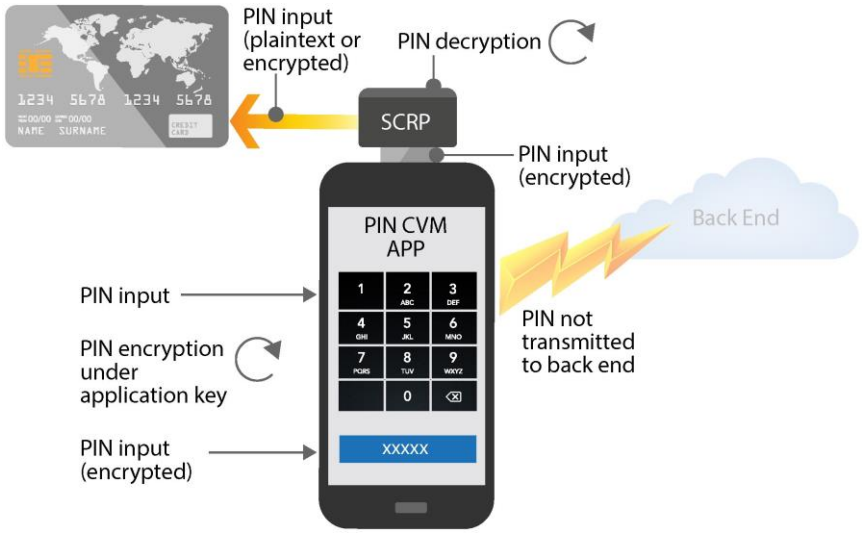


Figure 4:
Offline PIN Verification

- The SCRP performs offline PIN verification (contact EMV only).
- The SRED data is transmitted to an HSM within the back-end payment processing system.



Security Model – Traditional Hardware-based PIN Entry vs. Software-based PIN Entry

In a hardware-based PIN entry security model, the protection of the PIN is primarily the responsibility of the specialized PED hardware and software (including the firmware). Some hardware-based solutions may utilize back-end monitoring systems to supplement their security posture; however, traditional hardware PIN entry devices have all protections within the device itself. Detection mechanisms that identify when security controls are not in an approved state operate within the PED, and the response for the hardware security model is to disable the PED from processing PIN-based transactions and clear all sensitive data from the device. All PIN security protections are performed within the PED itself.

In a software-based PIN entry security model, the device where the PIN is entered and initially protected remains an important component. In addition, mechanisms that support attestation (to ensure the security mechanisms are intact and operational), detection (to notify when anomalies are present) and response (controls to alert and take action), play an equally significant role in the overall software-based PIN entry solution security model. Furthermore, having the device connected online provides opportunities to extend these capabilities to back-end monitoring systems.

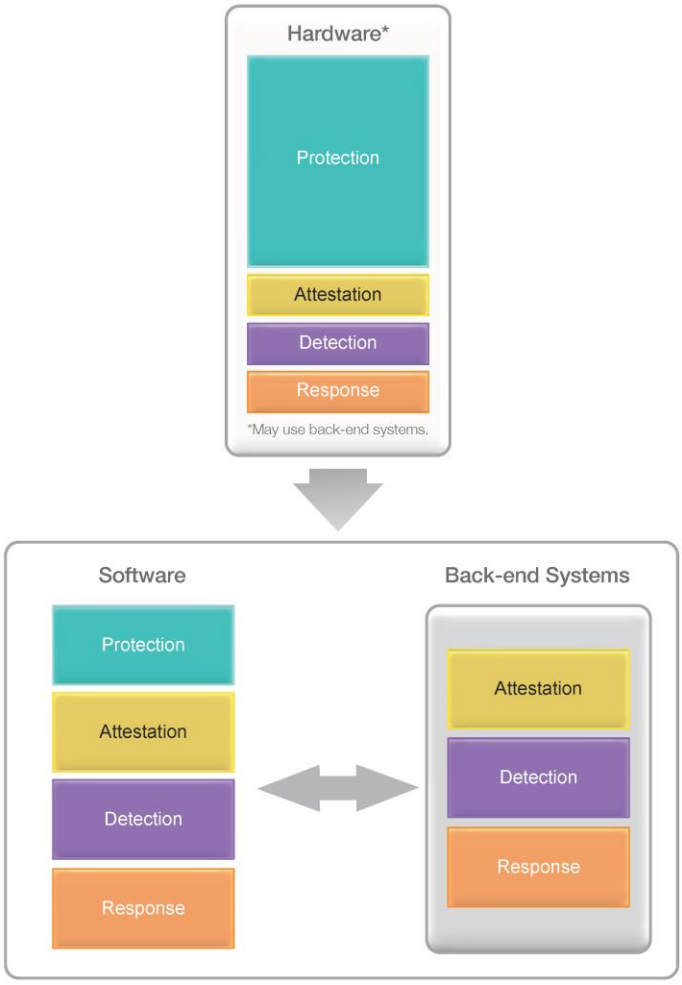


Figure 5: Hardware-based and Software-based PIN Entry

There are, however, individual components of a software solution where there is limited control—for example, the underlying mobile device hardware platform and Operating System (OS). Given that these are COTS devices, there is an assumption that these components (e.g., COTS OS, configuration of hardware components of a phone, etc.) are unknown or untrusted.

It must be assumed that an attacker has full access to the software that executes on any unknown or untrusted platform (where that "software" may be a binary executable, interpreted bytecode, etc., as it is loaded onto the platform). Therefore, it is considered important for the software to provide inherent protections that complicate reverse engineering and tampering of the code execution flow. This may include, but is not limited to, protections using "obfuscation" of the code, internal integrity checks for code and processing flows and encryption of code segments, etc.

Software-based PIN entry architecture relies upon the following components that combine to provide for protection, attestation, detection, and response controls:

1. An SCRP device that supports the solution. The SCRP is SRED-enabled and connected to the COTS device. The SCRP provides:
   a. Protection of cardholder data sourced from the payment instrument
   b. Decryption of encrypted PIN data received from the PIN CVM application
   c. Translation of PIN data into required PIN-block format (This would only apply to online PIN verification processing.)
   d. Re-encryption of PIN data
2. A PIN CVM application that resides on the COTS device and:
   a. Provides a secure User Interface (UI) for PIN entry
   b. Performs initial encryption of the PIN
   c. Passes attestation health-check data about the SCRP, COTS platform, and PIN CVM application to the monitoring environment
   d. Contains software protection mechanisms to maintain its own integrity against attack
   e. Delivers the encrypted PIN to the SCRP to be decrypted/re-encrypted for it to be passed to the back-end monitoring and attestation systems
3. A Commercial Off-The-Shelf (COTS) device that is operated by the merchant to run the PIN CVM application as well as the SCRP. The COTS may have a Trusted Execution Environment (TEE) built in but it is not a requirement.
4. A set of back-end systems that perform functions for the PIN CVM solution such as:
   - Attestation
     Processes attestation health-check data from the PIN CVM application and enforces pre-established security policies
   - Monitoring
     Monitors and provisions security controls to detect, alert and mitigate suspected or actual threats and attacks against the SCRP, PIN CVM application, and the COTS device
   - Processing
     Processing environment that receives encrypted cardholder and PIN data (for online PIN) from the SCRP

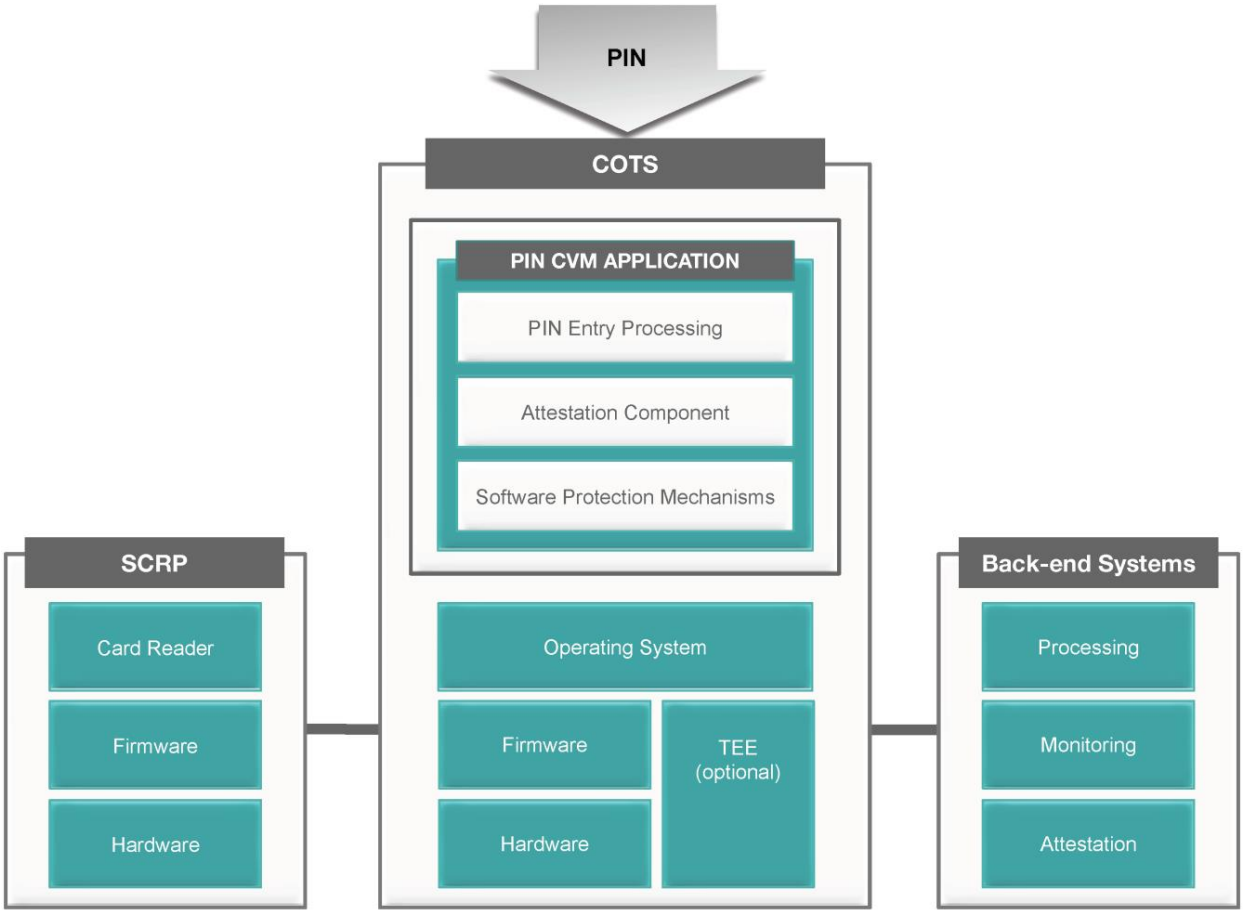


Figure 6: Example of PIN CVM Solution Architecture

Scope of the Target of Evaluation

The scope of the TOE, which will also then be assessed under this standard, are defined as the following:

1. Merchant payment processing application(s), referred to as the software PIN acceptance application(s), which are involved in PIN payment acceptance and/or may influence the security of payment data processing. These applications may be in part or as a whole executed on a COTS system or executed remotely and rendered on the COTS system through other means.
2. Secure Card Reader-PIN (SCRP), which is able to accept EMV-based payment cards. Optionally, in accordance to Software-based PIN Entry on COTS™ Magnetic Stripe Readers Annex, the SCRP may provide the physical interface necessary for reading magnetic stripe cards.
3. A back-end monitoring system (monitor) that cannot be entirely (logically) accessed from the COTS device and which must be capable of being regularly/rapidly updated to respond to new threats to the payment processing environment and changes or updates to the COTS solutions that may be used.
4. A back-end processing environment that performs the transaction processing. This system is treated as logically different from the monitor system but may be integrated with the monitor environment.
5. Assessment of the integration of the above, where the solution is not provided by a single vendor and/or integration of the components may lead to potential insecurities.

It is expected that the monitoring system will integrate both local and remote features, to allow for the identification of new types of attacks and rapid response and deployment of updated mitigations against such threats.

Organization of this document

This document is organized as follows:

1. Core Requirements
2. PIN Cardholder Verification Method (CVM) Application
3. Back-end Systems – Monitoring/Attestation
4. Solution Integration Requirements
5. Back-end Systems – Processing

Core requirements are items that may apply to all areas and components that are being assessed. For example, the random numbers requirement will apply to all parts of the TOE where random numbers are generated for security functions. These core requirements must therefore be assessed against all parts of the TOE – the PIN CVM application, SCRP, backend-end monitoring/attestation and back-end systems processing. However, they may be assessed separately, depending on the area(s) of the TOE that are currently under evaluation; if the PIN CVM application alone is being evaluated, only this will be assessed against the Core requirements. However, at some point all areas of the total solution must be assessed to each of the Core requirements.

PIN CVM application requirements and back-end monitoring and attestation system requirements apply only to those sections of the TOE and may be assessed against those components without considering the entire solution.

Solution integration requirements are a combination of requirements that must be assessed against the entire solution and parts of the solution that involve back-end payment processing and ownership of system risk analysis. It should be noted that some parts of these requirements presuppose functions (such as the creation and use of secure channels between disparate components) in parts of the solution that may be otherwise separately assessed (such as the monitoring system or PIN CVM Application).

Lastly, the back-end systems processing requirements apply only to solution components that perform decryption of cardholder account data and PIN.

Structure of the TRs

Each PCI requirement as stated in the *PCI Software-based PIN on COTS Test Requirements* is represented by a subsection. For example, Requirement A1 is represented in this document as:

TR A1 Protection of Sensitive Services

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 TRs.

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

For example, the first TR under A1 is:

TA1.1

Reporting Requirements for Software-based PIN Entry on COTS Laboratories

The evaluation report(s) must present evidence of a solution'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
- Includes information relevant to any applicable FAQs
- Conforms to Laboratory General Requirements and any other related documentation in the PTS Program, such as (but not restricted to) Reporting Guidance and Templates
- Conforms to documentation listed in the SPOC Program Guide

Minimal Contents of Reports and Minimal Test Activities

All reports shall include an overview and summary section at the beginning of the document. This summary shall include the following at a minimum. Refer to the SPOC Program Guide for additional information:

- A system overview that summarizes the design, hardware and software architectures, functionalities, and any other security-relevant attributes, features, or functions including (but not restricted to):
  - SCRP(s) that may be used with the system
  - Any back-end or "cloud-based" components that are used as part of the system
  - Test COTS devices that were used in the evaluation of the system
- A summary list of TRs with verdicts on whether tested and whether compliant
- A summary of any assistance provided by the vendor to the laboratory

In support of applicable test procedures, 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, analysis of white-box cryptographic implementations, etc.) to avoid prohibitively lengthy test activities.

The solution vendor shall make source code available to the lab and help make a systematic review of relevant security functions.

The evaluation report document shall demonstrate compliance to Security Requirements. For all TRs, 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 – i.e., considering TRs, FAQs, Program Guide and any other related documents. Every TR 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 TRs 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 system 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;
- Justifications for any reliance on test evidence not derived directly from the evaluation activities;
- Explanations for any conclusions based on theoretical analysis instead of applied testing.

The tester shall detail where conclusions or evidence is based on laboratory testing or vendor documentation/assertions and provide justification for any use of such vendor data rather than actual testing by the laboratory. Test evidence supplied by vendors that is used as the sole source of data for any requirement without any laboratory testing may result in the rejection of the report by PCI SSC.

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

In most cases, the TR 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 (such as an SCRP), relevant information clearly discernible in a graph, images capturing displays and other outputs, source code fragments, etc.

All TRs 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 SSC to identify test evidence following device approval.

Glossary

Term	Definition
Account data	Account data consists of cardholder data and/or sensitive authentication data.
AES	Abbreviation for “Advanced Encryption Standard.” Block cipher used in symmetric key cryptography adopted by NIST in November 2001 as *FIPS PUB 197* (or *FIPS 197*).
Ahead of Time (AoT) compilation	Compiling of code at some arbitrary time prior to the need to execute the code.
Asymmetric encryption	Also known as public key cryptography. A cryptographic technique that uses two related transformations, a public transformation (defined by the public key) and a private transformation (defined by the private key). The two transformations have the property that, given the public transformation, it is not computationally feasible to derive the private transformation. A system based on asymmetric cryptographic techniques can be an encipherment system, a signature system, a combined encipherment and signature system or a key-agreement system. With asymmetric cryptographic techniques, there are four elementary transformations: sign and verify for signature systems, and encipher and decipher for encipherment systems. The signature and the decipherment transformation are kept private by the owning entity, whereas the corresponding verification and encipherment transformations are published. There exist asymmetric cryptosystems (e.g., RSA) where the four elementary functions may be achieved by only two transformations: one private transformation suffices for both signing and decrypting messages, and one public transformation suffices for both verifying and encrypting messages. However, this does not conform to the principle of key separation and, where used, the four elementary transformations and the corresponding keys should be kept separate.
Attestation	The act of attestation in this standard is the interaction between a verifier (possibly server-based) and a prover (possibly client-based) to determine the current security state/behavior of the prover based on predefined measurements and thresholds provided by the prover.
Attestation system	The set of components that perform attestation processing for the PIN CVM solution. Its components include the PIN CVM application attestation component and the back-end attestation component—the latter works in close association with the back-end monitoring system.
Attestation component	An element of the PIN CVM solution that performs attestation processing.
Authentication	The process for unambiguously establishing the identity of an entity, process, organization or person.
Back-end systems	The set of systems providing the server-side functionality of the PIN CVM solution. These functionalities include monitoring, attestation, and transaction processing. In addition, the back-end systems include the IT environments necessary to support the functionalities of the PIN CVM solution.

Term	Definition
Cardholder	Customer to whom a payment card or card proxy is issued, or any individual authorized to use a payment card or card proxy.
Cardholder data	At a minimum, cardholder data consists of the full PAN. Cardholder data may also appear in the form of the full PAN plus any of the following: cardholder name, expiration date, and/or service code. See *sensitive authentication data* for additional data elements that may be transmitted or processed (but not stored) as part of a payment transaction.
Cardholder Verification Method (CVM)	A method of authenticating a cardholder during a transaction. Common CVMs include signature, PIN, and biometrics.
CDE	Acronym for "cardholder data environment." The people, processes, and technology that store, process, or transmit cardholder data or sensitive authentication data.
Clear text	Intelligible data that has meaning and can be read or acted upon without the application of decryption. Also known as plaintext.
Commercial Off-The-Shelf (COTS) device	A mobile device (e.g., smartphone or tablet) that is designed for mass-market distribution.
Consumer	Individual purchasing goods, services or both.
Compiling	Translation of computer code from one format into another format. Usually used to take human-readable "source" code and transform this into a format that can be executed by a specific platform or execution environment.
Correlatable data	In the context of this standard, this is data that would facilitate the correlation of a PIN with a separate transaction or database that contains cardholder data such that interception of this data and the entered PIN could reasonably lead to the association of the PIN with its PAN. Examples might include time and date stamps, device identifying information, and loyalty program identifiers.
COTS platform	The hardware of the COTS device.
Deterministic Random Number Generator (DRNG)	A deterministic algorithm that generates a sequence of numbers with little or no discernible pattern in the numbers, except for broad statistical properties, which uses a seed value provided by a Non-deterministic Random Number Generator.
Dual control	A process of using two or more separate entities (usually persons), operating in concert, to protect sensitive functions or information. Each entity is equally responsible for the physical protection of materials involved in vulnerable processes. No single person must be able to access or to use the materials (e.g., cryptographic key). For manual key generation, conveyance, loading, storage, and retrieval, dual control requires split knowledge of the key among the entities. No single person can gain control of a protected item or process. Also see *split knowledge*.

Term	Definition
ECC	Acronym for "Elliptic Curve Cryptography." Approach to public-key cryptography based on elliptic curves over finite fields.
EMV®	A payment standard that implements cryptographic authentication, published by EMVCo.
EMVCo	A privately owned corporation. The current members of EMVCo are JCB International, American Express, Mastercard, China UnionPay, Discover Financial, and Visa Inc.
Encryption	Process of converting information into an unintelligible form except to holders of a specific cryptographic key. Use of encryption protects information between the encryption process and the decryption process (the inverse of encryption) against unauthorized disclosure.
Endpoint	An interface exposed by a communicating entity or channel, i.e., a node originating or accepting a transaction session.
Environment	The IT environment supporting one or more functionalities of the PIN CVM solution—such as the IT environment hosting the back-end monitoring system.
Execution environment	The set of hardware and software on which a program is executed. This may be provided through hardware alone, include a combination of hardware and software elements, or be virtualized and implemented in software such that the execution environment can be similarly executed on different hardware platforms.
Full screen mode	Where the PIN CVM application that is currently executing is in control of the primary display and input mechanism(s) of the COTS device. A full screen mode may still include display features that are controlled and/or managed by the COTS operating system but may not include any display from other applications. It is assumed by this standard that full screen mode mitigates the use of any separately controlled or managed displays or input mechanisms to display prompts for data entry or capture such data entry.
Graphical User Interface (GUI)	A User Interface that is provided through images and text.
Hash	A (mathematical) function that is a non-secret algorithm, which takes any arbitrary-length message as input and produces a fixed-length hash result. Approved hash functions satisfy the following properties: a) One-way – It is computationally infeasible to find any input that maps to any pre-specified output. b) Collision-resistant – It is computationally infeasible to find any two distinct inputs (e.g., messages) that map to the same output. It may be used to reduce a potentially long message into a "hash value" or "message digest" that is sufficiently compact to be input into a digital-signature algorithm. A "good" hash is such that the results of applying the function to a (large) set of values in a given domain will be evenly (and randomly) distributed over a smaller range.

Term	Definition
Hash-based Message Authentication Code (HMAC)	A message authentication code that is produced using hash algorithms rather than a symmetric cryptographic algorithm. Defined in *FIPS 198-1.*
Human Machine Interface (HMI)	Human Machine Interface allows for the physical entry and digitization of a PIN. This may consist of a mechanical or "touch" keypad or an integrated sensor but cannot be solely an electronic interface without any physical interface for the customer to physically enter the PIN.
Integrity	Ensuring consistency of data; in particular, preventing unauthorized and undetected creation, alteration or destruction of data.
Just-In-Time (JIT) compilation	Compiling of code immediately prior to the execution of that code.
Key agreement	A key-establishment protocol for establishing a shared secret key between entities in such a way that neither of them can predetermine the value of that key. That is, the secret key is a function of information contributed by two or more participants.
Key generation	Creation of a cryptographic key either from a Random Number Generator or through a one-way process utilizing another cryptographic key. ***Note:*** *Key variants are not allowed.*
Key installation	Loading of a key that is protected with white-box cryptography, usually embedded within an application.
Key loading	Process by which a key is manually or electronically transferred into a secure cryptographic device.
Key management	The activities involving the handling of cryptographic keys and other related security parameters (e.g., initialization vectors, counters) during the entire life cycle of the keys, including their generation, storage, distribution, loading, and use, deletion, destruction, and archiving.
Key variant	A new key formed by a process (which need not be secret) with the original key, such that one or more of the non-parity bits of the new key differ from the corresponding bits of the original key.
Key Check Value (KCV)	A value used to identify a key without revealing any bits of the actual key itself. Check values are computed by encrypting an all-zero 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 TDEA and 5 bytes AES). This method may be used for TDEA. TDEA may optionally use, and AES uses a technique where the KCV is calculated by MACing an all-zero 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. A TDEA key or a component of a TDEA key will be MAC'd using the TDEA block cipher, while a 128-bit AES key or component will be MAC'd using the AES-128 block cipher. Also known as Key verification check (KVC).

Term	Definition
Key wrapping	A format for storage and transmission of symmetric cryptographic keys that embeds metadata about the key type and use, as well as providing cryptographic authentication across the encrypted key and this metadata to ensure that the key and its purpose cannot be altered.
Major OS version	For the purposes of this standard, a major OS version is considered to be a version that is available for use on multiple platforms/devices without further customization. For example, if a developer or device manufacturer is able to download source code from a code repository and without further customization compile this code (perhaps with the use or facility of additional drivers and hardware abstraction layers) for use on a device, that is considered a "major" OS version.
Man-In-The-Middle (MITM) attack	An attack method where a malicious third party interposes between two other communicating parties and modifies the data sent between them.
Mandatory access control	Access control by which the operating system constrains the ability of a process or thread to access or perform an operation on objects or targets such as files, directories, TCP/UDP ports, shared memory segments, IO devices, etc., though an authorization rule enforced by the operating system kernel.
Manual key loading	Loading of a cryptographic key using two or more full-length components or use m-of-n shares, entered directly through a secure physical mechanism.
MAC	In cryptography, an acronym for "message authentication code." A small piece of information used to authenticate a message.
Minor OS version	A minor OS version is where further customization has been performed by the device manufacturer or distributor, such that the operating system has been modified beyond the major version. A minor OS version includes only modifications where the underlying operations have not been modified, and not modifications that are made solely at the "application" level – such as simple "skinning" of a common operating system base so it appears unique.
Mobile device	In the context of this standard, see COTS device.
M of N	An m-of-n scheme is a component or share allocation scheme where m is the number of shares or components necessary to form the key, and n is the number of the total set of shares or components related to the key. Management of the shares or components must be sufficient to ensure that no one person can gain access to enough of the item to form the key alone.
Monitor	In this standard, the "monitor" is an implementation that may be shared across different execution environments, which provides a level of validation and assurance of the execution environment in which the PIN CVM application executes, providing a level of software-based tamper detection and response. The monitor must be capable of being regularly updated to detect and respond to new threats.

Term	Definition
Non-deterministic Random Number Generator (NRNG)	A Random Number Generator that has access to an entropy source and (when working properly) produces output numbers (or bit strings) that have full entropy. Sometimes called a True Random Number (or Bit) Generator. Contrast with a Deterministic Random Number Generator.
Obfuscation	Protection applied to a process or data through increasing the complexity of interpreting that data. For the purposes of this standard, "obfuscation" refers to "code obfuscation," where computational processes have been applied to increase the complexity of a code set to reduce the ability to reverse-engineer that code.
Offline payment transaction	In an offline EMV transaction, the card and terminal communicate and use issuer-defined risk parameters that are set in the card to determine whether the transaction can be authorized. Offline transactions are used when terminals do not have online connectivity—e.g., at a ticket kiosk—or in countries where telecommunications costs are high.
Offline PIN verification	A method of PIN verification that sends the PIN entered by the cardholder to the EMV chip on the card.
Online PIN verification	A process used to verify the cardholder's PIN by sending an encrypted PIN value to the issuer or its agent for validation in an authorization request.
Operating System (OS)	System software that manages the underlying hardware and software resources and provides common services for programs. Common operating systems in a COTS environment include, but are not limited to, Android and iOS.
OS store	A digital distribution service operated by the COTS OS vendor or by the COTS device manufacturer.
PCI DSS	The Data Security Standard published and maintained by the Payment Card Industry Security Standards Council. PCI DSS provides a baseline of technical and operational requirements designed to protect account data.
PCI PIN	A PCI standard that contains a complete set of requirements for the secure management, processing and transmission of Personal Identification Number data during online and offline payment card transaction processing at automated teller machines (ATMs) and attended and unattended point-of-sale (POS) terminals.
Personal Identification Number (PIN)	A numeric personal identification code that authenticates a cardholder. A PIN consists only of decimal digits.
Physical Unclonable Function (PUF)	An intrinsic value or transformation that can be provided by a system that is a function of some physical process, such that it cannot be replicated or altered.
PIN block	Defined formats used for offline and online PIN processing and transmission, as defined in ISO 9564 Part 1.

Term	Definition
PIN CVM application	All parts of the code, regardless of execution environment, that are installed and executed on the merchant COTS device for the purposes of accepting and processing the cardholder's PIN. The client-side monitor and/or a payment application may be incorporated into the PIN CVM Application or may be a separate application.
PIN CVM solution ("the solution")	The set of components and processes that support the entry of PIN data into a COTS device. At a minimum, the solution includes SCRP, PIN CVM application, and the back-end systems and environments that perform attestation, monitoring and payment and online PIN processing.
Private key	A cryptographic key used with a public-key cryptographic algorithm that is uniquely associated with an entity and is not made public. In the case of an asymmetric signature system, the private key defines the signature transformation. In the case of an asymmetric encipherment system, the private key defines the decipherment transformation.
Public key	A cryptographic key used with a public-key cryptographic algorithm that is uniquely associated with an entity and may be made public. In the case of an asymmetric signature system, the public key defines the verification transformation. In the case of an asymmetric encipherment system, the public key defines the encipherment transformation. A key that is "publicly known" is not necessarily globally available. The key may only be available to all members of a pre-specified group.
Public key cryptography	See *asymmetric encryption*.
Random Number Generator (RNG)	The process of generating values with a high level of entropy and that satisfy various qualifications, using cryptographic and hardware-based "noise" mechanisms. This results in a value in a set that has equal probability of being selected from the total population of possibilities, hence unpredictable.
Replay attack	A replay attack (also known as playback attack) is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed.
RSA	An asymmetric encryption algorithm that was defined by the cryptographers Rivest, Shamir, and Aldeman.
Secure boot	See *trusted boot.*
Secure Card Reader – PIN (SCRP)	A physical card reader that has been assessed compliant to the PCI PTS SCRP Approval Class and is listed on the PTS approval website.
Secure channel	A cryptographically protected connection between two processing elements.
Secure Cryptographic Device (SCD)	A physically and logically protected hardware device that provides a secure set of cryptographic services. It includes the set of hardware, firmware, software or some combination thereof that implements cryptographic logic, cryptographic processes or both, including cryptographic algorithms. Examples include ANSI X9.24 part 1 or ISO 13491.

Term	Definition
Secure Reading and Exchange of Data (SRED)	Requirements contained in the PCI PTS POI Standard, detailing the security controls for devices that protect account data.
Sensitive authentication data	Security-related information—including but not limited to card validation codes/values, full track data (from the magnetic stripe or equivalent on a chip), PINs and PIN blocks—used to authenticate cardholders and/or authorize payment card transactions.
Sensitive data	For the purposes of this standard, sensitive data is cryptographic materials—e.g., keys, certificates, cardholder PINs, or cardholder data.
Sensitive services	Those functions that affect underlying processes that support the protection of sensitive data—e.g., cryptographic keys, PINs and cardholder data.
Split knowledge	A condition under which two or more entities separately have key components or key shares that individually convey no knowledge of the resultant cryptographic key. The information needed to perform a process such as key formation is split among two or more people. No individual has enough information to gain knowledge of any part of the actual key that is formed.
Software protection mechanisms	Methods and implementations used to prevent the reverse engineering and modification of software. See *obfuscation* and *white-box cryptography* as examples of commonly used software protection mechanisms.
Solution provider	An entity that develops, manages and/or deploys PIN CVM solutions.
Supported platform	The current operating system supported by the operating system vendor.
Symmetric encryption	A cryptographic key that is used in symmetric cryptographic algorithms. The same symmetric key that is used for encryption is also used for decryption. Also known as “secret key.”
Tamper detection	The automatic determination by a cryptographic module that an attempt has been made to compromise the security of the module.
Tamper responsive	A characteristic that provides an active response to the detection of an attack, thereby preventing a success.
TDES	An algorithm specified in *ISO/IEC 18033-3: Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers. (PIN)*
Test Requirements (TR)	Requirements that dictate the set of tests that must be performed to confirm compliance with a specific standard.
The solution	See *PIN CVM solution.*
Third-party app stores	App stores that are not supported by the COTS OS vendor and are not pre-installed by the device manufacturer.
True Random Number Generator (TRNG)	A device that generates random numbers from a physical process, such as a Physical Unclonable Function, rather than a deterministic algorithm.

Term	Definition
Trusted boot	A cryptographic process where the bootloader verifies the integrity of all components (e.g., kernel objects) loaded during OS start-up process, before loading. Also known as verified boot and secure boot (e.g., Google or Apple).
Trusted Execution Environment (TEE)	A Trusted Execution Environment provides security features such as isolated execution environment for trusted applications ("trustlets"). It protects security assets from general software attacks, defines safeguards as to data and functions that a program can access and resists a set of defined threats.
User Interface (UI)	The set of the human-machine interfaces that allows for interaction between a person and a computerized system.
White-box cryptography	A method used to obfuscate a cryptographic algorithm and key with the intent that the determination of the key value is computationally complex.

Publications and References

PCI SSC Standards https://www.pcisecuritystandards.org/document_library	
DSS	Data Security Standard
DESV	Designated Entities Supplemental Validation (Appendix A3 in PCI DSS v3.2)
HSM	PIN Transaction Security (PTS) Hardware Security Module Security Requirements
PIN	PIN Security Requirements
POI	PTS Point of Interaction Modular Security Requirements

Other Industry Security References	
ANSI X9.24	Retail Financial Services Key Management
FIPS 140-2	Federal Information Processing Standard, Security Requirements for Cryptographic Modules
FIPS 186-4	Federal Information Processing Standard, Digital Signature Standard (DSS)
FIPS 198-1	Federal Information Processing Standard, The Keyed-Hash Message Authentication Code, (HMAC)
FIPS PUB 197 (or "FIPS 197")	Federal Information Processing Standards Publication 197 ADVANCED ENCRYPTION STANDARD (AES)
ISO 7816	Identification cards -- Integrated circuit cards
ISO 9564-2	Financial services -- Personal Identification Number (PIN) management and security -- Part 2: Approved algorithms for PIN encipherment
ISO 9564-1	Financial services -- Personal Identification Number (PIN) management and security -- Part 1: Basic principles and requirements for PINs in card-based systems
ISO 14443	Identification cards -- Contactless integrated circuit cards -- Proximity cards
ISO 11568	Financial Services, Key Management
ISO 13491	Financial Services, Secure Cryptographic Devices
ISO 16609	Banking Requirements for message authentication using symmetric techniques
ISO/IEC 18033-3	Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers
NIST SP 800-22	National Institute of Standards and Technology, Special Report on A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications

Other Industry Security References	
NIST SP 800-38B	National Institute of Standards and Technology, Special Report on Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication

TR Module 1: Core Requirements (to be applied to all in-scope areas)

TR A1: Protection of Sensitive Services

***Sensitive services** are those functions that affect underlying processes that support the protection of **sensitive data**—e.g., cryptographic keys, **PIN**s and **cardholder data**. All **sensitive services** must be identified and support the confidentiality, **integrity** and availability of **the solution**. Entering or exiting **sensitive services** must not reveal or otherwise affect **sensitive data**.*

Guidance

A sensitive service is any service that may affect the security of the overall system. Common examples are key loading, modification or update of monitor services and cryptographic signing of assets to allow their authenticity to be verified.

All sensitive services must be performed under dual control or using systems and cryptographic keys that have been installed and are managed under dual control (e.g., Cryptographic keys within an HSM). In this context, dual control implies that at no point in the process can a single person be responsible for the security (or compromise of the security) of the system.

Operations that involve secret or private cryptographic keys must be performed using split knowledge. Split knowledge requires that no one person is able to determine any single bit of a secret or cryptographic key. Split knowledge can be provided by storing keys on SCDs that will not output the plaintext key, or through use of two or more full-length components during key loading or a valid m-of-n secret-sharing scheme. Use of smartcards for storing full cryptographic keys is not considered compliant because a smartcard is not tamper responsive, and therefore is not considered an SCD. However, smartcards may be used to store individual components (no more than one component per smartcard).

TA1.1 The tester shall confirm that vendor documentation exists, is followed and updated at least annually and details all sensitive services implemented by the system. At a minimum, this shall include key loading (for all in-scope areas), signing of firmware and applications and modifications to the monitor functionality or configuration.

TA1.2 Referencing the key table produced in TR A4: Key Management, the tester shall provide details on each of the key-loading methods and procedures used in the system. This will include, but may not be limited to, loading of keys into back-end HSMs, use of white-box cryptography keys, loading of keys into the SCRP and initial provisioning of keys into the PIN CVM application. The creation of white-box keys is an exception to the requirements in this TR.

TA1.3 For each of the above key-loading methods, the tester shall confirm that it enforces dual control across the entire process, such that no one person is ever solely responsible (or can be held liable) for the security of the cryptographic key-loading or key-injection operation.

TA1.4 Where any of the above key-loading methods involve secret or private cryptographic keys, the tester shall confirm that the process ensures split knowledge, such that no one person is ever able to determine any single bit of the actual cryptographic key.

TA1.5 The tester shall confirm how modifications and/or updates to the monitoring system are performed and confirm that this requires dual control prior to the update "going live" in the production environment and any changes follow a formal change-control procedure.

TA1.6 The tester shall confirm how the PIN CVM application code, or any other security related assets, is/are signed, and confirm that this involves a dual controlled process using cryptographic keys that are maintained in one of the approved forms (as defined in TA4.3). An exception to this item is permissible only where the PIN CVM application signature is required to be generated by a third party (such as an OS store) outside the control of the PIN CVM application vendor.

TA1.7 For PIN CVM applications that rely in part on security provided by a TEE, the tester shall confirm that any code or data loaded into the TEE requires a signature that is created using procedures that are compliant to the requirements for a sensitive service.

TA1.8 For any other sensitive services, the tester shall confirm how dual control is maintained.

TR A2: Random Numbers

Random numbers must be generated using a process that ensures sufficient entropy (as defined in NIST SP 800-90b) and lack of statistical correlation. Random number generation on the PIN CVM application must always include a random seed provided by the SCRP, passed through a DRNG on the COTS device.

Guidance

Random numbers are relied upon by many security processes and secure communications methods. Generation of random numbers with insufficient entropy has been the cause of many high-profile vulnerabilities, and therefore the quality of the random numbers generated by the system is vital. Because this standard allows for the use of COTS devices that may not have a good source of entropy, 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 NRNG on an external SCRP (which has been validated through PCI PTS testing).

Random numbers that are not directly relied upon for security of the customer PIN, cardholder data or monitor/attestation data - e.g., random values used in TLS sessions, where the data being transmitted is otherwise protected using application level cryptography - do not need to meet this requirement.

TA2.1 The tester shall detail the implementation of all random number generation functions used in the implementation. This must include random numbers generated on all platforms supported by the solution (assessed under TR B12: Supported Platforms), as well as random numbers generated in any back-end systems (where these random numbers are used for the purposes of providing security to the system). The intent is not to check the NRNG process used by the OS of each supported platform, but to ensure that the PIN CVM application implements methods to ensure robust random data generation where there is the potential for supported platforms to have inherently poor random data processes of their own.

These details shall include any seed values used, hardware systems and software-based DRNGs.

TA2.2 The tester shall confirm that generation of random numbers on the PIN CVM application requires input of random data (of at least 256 bits) from the SCRP using its NRNG approved through PCI PTS testing. The tester shall ensure that initializing and/or seeding of the DRNG from the SCRP in this way cannot be abused to intentionally reproduce a given random value or sequence.

TA2.3 The tester shall list all security services implemented within the system that require or rely upon the use of random data. This may include generation of padding data (for example, for use in certificates, key bundles, EMV Unpredictable Numbers or offline encrypted PIN blocks), generation of cryptographic keys, randomization of the keypad button layout, etc.

TA2.4 The tester shall review the source code of each of these services and confirm that they correctly utilize the RNG 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.

TA2.5 The tester shall confirm that the NRNG implementation in the back-end monitoring system environment is implemented using a random number implementation approved to FIPS140-2 L3 or PCI PTS approved HSM. Where approval to FIPS140-2 L3 is used, the tester shall validate from the approval that the entropy is not externally supplied.

TA2.6 The tester shall confirm that the PIN CVM application implements a software-based DRNG for all random data operations and confirm that this DRNG implements cryptographic algorithms (which may include hash algorithms) to whiten any random data sources used for the seed values. This DRNG must be implemented within the PIN CVM application. Reliance entirely upon random data provided by the base OS is non-compliant to this requirement.

TA2.7 The tester shall obtain at least 128MB of random data from each of the DRNG implementations used in the system. This data may be supplied directly by the vendor for testing. The tester shall detail the method used to generate this data and justify why this sufficiently replicates the way in which the DRNG will be used by the system. The tester shall pass the 128MB of data through the NIST STS test program (as defined in NIST SP 800-90b) and detail the results, indicating pass and fail results and how these demonstrate compliance to this TR. In some situations, it is necessary to repeat such tests using additionally obtained data to confirm final results.

TR A3: Acceptable Cryptography

The system shall use only industry-standard cryptographic algorithms for any security services. Cryptographic algorithms used for security services shall use only those algorithms and minimum key lengths as outlined in Appendix C: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms.

Guidance

The PIN must be encrypted on the COTS device for transport to the SCRP, where it is then translated under a cryptographic key shared between the SCRP and the back-end systems .

Acceptable hash functions use SHA256 or above. This does not preclude the use of other hash types where collision resistance is not required as a security feature (e.g., fingerprinting or file comparison).

Encryption used to protect PIN or tamper-detection data must be performed using a key that is unique per transaction or communication session and per PIN CVM application instance.

Customer PINs transmitted from the SCRP to the PCI PIN-compliant processing host may be encrypted using TDES as the only exception to these requirements.

TA3.1 The tester shall provide a list of all cryptographic operations used by the system for security services and note the cryptographic algorithm used for each of these cryptographic operations. This must include (but not be limited to) any cipher suites or other cryptographic algorithms implemented within transport layer protocols that are used for secure channels (as assessed under TR D2: Secure Channels).

TA3.2 Given the above, the tester shall confirm that these algorithms meet the minimum requirements outlined in Appendix C: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms. Where key derivation methods are used for creating the transaction unique key, the tester shall verify that the method is not reversible and provides perfect forward secrecy.

TA3.3 The tester shall confirm that the PIN CVM application and SCRP ensure that PIN data is encrypted with a unique key for each transaction, and detail the methods used to generate this key.

TA3.4 The tester shall confirm that the monitoring system ensures that any tamper-detection or response messages are communicated between the COTS and back-end monitoring systems using a unique key per session.

TA3.5 The tester shall confirm that security services provided by the supported platforms (assessed under TR B12: Supported Platforms) implement strong cryptography as defined under this requirement. Where exceptions exist, the tester shall justify how the overall solution mitigates these exceptions.

TR A4: Key Management

The key-management techniques implemented in the solution conform to ISO 11568 and/or ANSI X9.24. Key-management techniques must implement methods such as key wrapping to protect the integrity and purpose of symmetric cryptographic keys. Public keys must be stored in a manner that ensures their authenticity.

Guidance

This requirement applies to components of the solution, including back-end components, the PIN CVM application, the monitoring system and the SCRP. Where software encryption is used—e.g., white-box cryptography—key-management practices must adhere to requirements specified in Module 2: PIN CVM Application Requirements.

Secret and private cryptographic keys that are relied upon for security must be unique per device/application. Shared public keys are acceptable, but methods and procedures to revoke compromised public/private key pairs must be implemented.

Additional information on PKI can be found in X9.79-4.

TA4.1 The tester shall provide a separate "key table" for each of the in-scope areas of the current assessment – i.e., relevant PIN CVM application(s), SCRP, back-end monitoring/attestation system – that outlines all cryptographic keys used in the system that are relied upon for the security of the PIN or overall system security. This key table shall have at least the following columns: key name, key purpose, key size, key location – e.g., device/system/memory location, as appropriate – algorithm, method used to load/generate key, whether the key is unique to transaction/device/overall system (or other), how often the key is changed, how the key is secured – e.g., in an HSM, in the SCRP, using white-box cryptography, by encryption with another key, etc. – how the key is erased/destroyed. An example of a blank key table is provided below:

Key name	Key purpose	Key size	Key location	Algorithm	How loaded/ generated/ exported	Unique per	Crypto period	Secured by	Erased by

TA4.2 The tester shall detail any key-generation or key-agreement processes that are used by the system and confirm that they ensure that keys are generated with equivalent entropy – e.g., a 128-bit key is generated with 128 bits of entropy input.

TA4.3 Where padding is used prior to/during encryption methods, the tester shall detail the padding methodology implemented and confirm that asymmetric encryption always implements industry-standard padding methods.

TA4.4 The tester shall confirm that the key-loading processes ensure that secret or private cryptographic keys are always maintained in one of the following approved forms:

- Encrypted by a key of equal or greater strength
- Stored within a SCD
- Managed as two or more full-length components
- Managed under a valid m-of-n secret-sharing scheme

TA4.5 The tester shall confirm that no reversible key calculation modes (such as key variants) are used to directly create new keys from an existing key. All key-generation functions must implement One-way Functions or other irreversible key-generation processes.

TA4.6 The tester shall confirm that any key "signature" or "fingerprint" values returned by the system do not reveal any details about the key itself. KCVs must be limited to 5 bytes or less, and hash algorithms used for key fingerprints (on secret or private keys) must implement SHA256 or above.

TA4.7 The tester shall determine from vendor documentation what the "crypto period" is for each key, how each key is updated and the old value destroyed at the end of this crypto period.

TA4.8 The tester shall confirm that security is not provided to any key by a key of lesser strength (e.g., by encrypting a 256-bit AES key with a 128-bit AES key).

TA4.9 For any public keys used by the system, the tester shall confirm how the authenticity of that public key is maintained. Use of public keys that are not signed or MAC'd, or that are maintained in self-signed certificates, is prohibited unless the authenticity of the key is ensured through use of an SCD instead. Self-signed certificates that exist as part of the base COTS platform on which the PIN CVM application is executed are excluded from this requirement. However, the self-signed certificates must not be relied upon for security services by the PIN CVM application unless the integrity and authenticity of the key is ensured.

TA4.10 The tester shall confirm how key purpose and integrity is ensured for all keys used in the system, preventing a key of one purpose (e.g., PIN encryption) from being replaced with a key of another purpose (e.g., general data encryption).

TA4.11 The tester shall confirm that each key has a single unique purpose and that no keys are used for multiple purposes (such as both signing and encrypting data), and that keys used to encrypt PIN data are not used for any other operation (such as general-purpose data encryption or monitor message encryption, etc.).

TA4.12 Keys used to validate the authenticity of a datagram must be unique to each endpoint, so that a(n) (H)MAC or signature generated at one end would always be different if generated by the other endpoint.

TA4.13 The tester shall confirm that, with the exception of any white-box keys, secret cryptographic keys are stored and maintained using key wrapping techniques.

TA4.14 The tester shall confirm that, with the exception of any white-box keys, all secret and private keys are unique to each COTS device, PIN CVM application instantiation or SCRP. Use of a single asymmetric key pair across multiple devices is acceptable where the public key resides on one or more COTS devices and the private key is secured within an HSM on the back-end system.

TA4.15 The tester shall note where public or white-box keys are not unique per COTS device, PIN CVM application instantiation or SCRP and confirm that methods and procedures to revoke and/or replace such keys (or key pairs) if compromised exist.

TA4.16 The tester shall detail how any white-box keys are generated, such that they are ensured to have sufficient entropy and the key value is not exposed during the generation process. Generation of white-box keys must use entropy from an approved RNG, as assessed under this standard. Where the white-box key-generation process is implemented outside a SCD, this device must be:

- Standalone (e.g., without modems, not connected to a LAN or WAN, not capable of wireless connections, etc.);
- Dedicated to only the white-box generation function (i.e., there must not be any other application software installed); and
- Located in a physically secure room that is dedicated to the white-box generation activities.

TA4.17 The tester shall verify that a mechanism exists for generation of audit logs for all key-management activities and all activities involving clear-text key components. The tester shall verify the audit log mechanism includes integrity protections against unauthorized modifications or deletions.

TA4.18 The tester shall verify that incident response procedures exist.

TA4.19 The tester shall confirm that cryptographic keys to which customer PINs are translated within the SCRP are controlled by the payment acquirer and generated/managed within the PCI PIN-compliant environment (as assessed under TR E2: PCI PIN Compliance).

TR A5: Secure Software Development Practices

System software shall be developed according to a defined software security development process. Developers must be demonstrably trained in security best practice for the platforms and languages used for security sensitive operations.

Guidance

Development of secure software requires knowledge of common attack techniques and vulnerabilities. These vulnerabilities can change over time, therefore a process to ensure continual update of the knowledge of the system programmers is vital. For compliance to this requirement, it is not sufficient to confirm that the programmers have been provided with documentation, books and/or training on secure software development. There must be auditable confirmation that the developers have knowledge of common vulnerabilities in the language and environment on which they develop the applications.

This requirement covers the development of software for all aspects of the solution, including the PIN CVM application, SCRP code and back-end monitoring system.

TA5.1 The tester shall confirm that all software is developed according to the development process and to the development requirements outlined in the *PCI Software-based PIN Entry on COTS Security Requirements,* Appendix D – Application Security Requirements.

TR A6: Security Testing

A penetration test must be performed on the system prior to initial deployment and at least once per year thereafter, and a security-flaw-reporting program must be implemented to encourage the finding and reporting of vulnerabilities by internal and external entities.

Guidance

Processes for detecting security vulnerabilities must include confirming the security of the system baseline, as well as any vendor-developed code. Penetration tests may be performed by internal staff, but any penetration testers must be able to demonstrate skill and knowledge in the art through formal accreditation to standards such as CREST and/or OSCP. System vendors must have an active vulnerability-reporting and management program that is commensurate with industry best practice and must be able to demonstrate remediation of security vulnerabilities reported through such public programs.

The scope of any vulnerability-reporting programs must include all consumer-accessible systems, including the PIN CVM application itself as well as the protocols used to communicate between the PIN CVM application, SCRP and back-end monitoring systems.

Penetration testing must include the PIN CVM application and back-end monitoring environment.

The requirement to institute a vulnerability-management program does not replace the requirement to perform penetration testing. Both must be implemented for compliance to this TR.

TA6.1 The tester shall confirm that if a vulnerability-reporting program exists for the system, there is evidence of accepting and remediating security vulnerabilities found through this program. The vendor must be able to demonstrate that any such vulnerabilities are processed through its risk and update process and patched accordingly.

TA6.2 The tester shall confirm that the vendor has a documented process that is demonstrably in use for the discovery and remediation of security vulnerabilities in its system.

TA6.3 The tester shall confirm that a penetration test has been performed on the system. The scope of the penetration test(s) must include all devices and OSs within the system baseline. The tester shall further confirm that a documented policy exists to require the repeat of the penetration test at least every year. Where sampling is performed, the tester shall validate the sampling is appropriate.

TA6.4 The tester shall confirm that the vendor has an explicit procedure in place for the acceptance and processing of new vulnerabilities through external communications. This requirement does not mandate the implementation of a "bug bounty" program but does require that all reported vulnerabilities are formally registered and processed according to a documented process.

TA6.5 The tester shall confirm that there is a public-facing procedure for the reporting of vulnerabilities in the solution. This procedure must implement methods to ensure the confidentiality of the vulnerability as it is reported (e.g., a process that requires the reporting of a vulnerability to a shared "info@[company]" e-mail address, without additional encryption, would be non-compliant to this requirement). Use of a specific web portal secured with TLS (using acceptable cipher suites) and/or e-mails secured with strong cryptography are examples of acceptable methods to secure the confidentiality of vulnerability reporting.

TA6.6 The tester shall obtain a list of vulnerabilities reported through the vulnerability-management program and penetration testing and validate that these have been addressed as per the vendor's documented vulnerability-management program.

TR Module 2: PIN CVM Application Requirements

TR B1: PIN CVM Application Tamper Resistance

The PIN CVM application accepting the PIN input must be designed and implemented to resist reverse engineering, modification or other attempts to maliciously alter and influence the secure operation of the application code.

Guidance

It must be assumed that an attacker has full access to the software that executes on any unknown or untrusted platform (where that "software" may be a binary executable, interpreted bytecode, etc., as it is loaded onto the platform). Because of this, it is necessary for the software to provide inherent protections that complicate reverse engineering of and tampering with the code execution flow. This may include, but is not limited to, protections using "obfuscation" of the code, internal integrity checks for code and processing flows and encryption of code segments.

Obfuscation must reduce the efficacy of common tools that may be used for decompilation of the code. Obfuscation methods may include, but not be limited to, control-flow and data obfuscation, execution of code sections in remote/cloud environments, API renaming, etc. Where the application is provided as a number of files (i.e., libraries) note what protections are provided for the calls between the libraries.

These protections are not required across all code, but must be implemented to protect all code that provides PIN CVM application security features, such that code complexity can be demonstrated to be significantly increased, or execution can be shown to be possible only on un-modified environments - e.g., through use of a device Physically Unclonable Function to encrypt data/execution, or by implementing the PIN application code in a trustlet that can be executed only in a secure TEE.

This requirement is not intended to cover any tamper-detection or response features that may be provided by the combination of the PIN CVM application and the system monitor, or inherently provided by the platform itself (e.g., a tamper-responsive SCRP or when the platform on which the PIN CVM application executes provides some forms of tamper-responsive features). Such requirements are covered under TR C2: System Tamper Response. This B1 requirement covers only tamper-resistance features that are wholly integrated and incorporated into the application and COTS device itself.

The tester is required to develop costings for the most feasible attack for this requirement. Although minimum costing point values are not specified in this version of the standard, the quality and efficacy of the monitoring system is considered vital for the security of the overall Software-based PIN Entry on COTS solution. Therefore, it is to be considered a non-compliance if values of "low" are assigned to any of the monitor items in the costing produced.

TB1.1 The tester shall review documentation and other evidence provided by the application developer and using this information shall detail the protections provided to the application to protect against tampering and reverse engineering, as attested by the vendor.

TB1.2 The tester shall provide details of all areas where functions provided by the application are executed. This will include the main processing environment of the COTS device but may also include other local execution environments (such as a TEE or embedded security processor).

TB1.3 The tester shall outline all areas of the PIN CVM application that are protected by the tamper-resistance measures. This must include the PIN handling code, SCRP/application/monitor interface code, memory and storage and any code that is involved in the use or security of cryptographic keys (both public and private/secret keys).

TB1.4 Where cryptography is applied to the code for the purposes of obfuscation and anti-tamper, the tester shall note what areas are protected, what protection is provided by the cryptography (confidentiality, integrity or both) and what algorithms are used, and confirm that these meet the requirements of TR A3: Acceptable Cryptography. The tester shall further note how any keys used for these cryptographic operations are protected. Any such keys must be referenced in the key table of this report (see TR A4: Key Management). Key loading is considered separately in TR A1: Protection of Sensitive Services.

TB1.5 Where protections are provided (partially or wholly) through code obfuscation, the tester shall:

- Examine provided application installation files where the protection methods have been applied, and through review of these (and comparison to files where protections have not yet been applied) comment on the validity of the vendor attestations and documentation regarding protection methods implemented
- Comment on the comparative file sizes between unprotected and protected application samples, as well as the relative compression ratio of each file type when standard compression functions are applied (directly to each obfuscated code segment)
- Attempt extraction of data objects (such as ASCII strings and symbols) and functional linking/interface tables (such as PLT/GOT), and note any differences between code sets before and after the obfuscation process
- Analyze and comment on the comparative code flow and linkage between the code before and after the obfuscation process
- Note and comment on any areas of non-traditional execution, where the obfuscation relies on virtualized/interpreted commands, non-deterministic operations or polymorphic processes, etc. It is expected that the tamper-resistance features applied will use one or more of such techniques and that relying purely on the obfuscation of the standard code execution flow will not be sufficient to meet these requirements.

The intent of this test item is for the tester to detail the security features of the obfuscation as implemented within the code of the PIN CVM application, so that this information may be used during the analysis attempt to break the obfuscation.

TB1.6 Where protections are partially provided by the operating environment, the tester shall:

- Confirm that the supported platforms (as assessed in TR B12: Supported Platforms) provide the required tamper-resistance features and can be used for the PIN CVM application
- Detail the protections that are provided by the operating environment and confirm how they provide the required tamper-resistance features as well as whether the protections can be disabled or deactivated by the user or application(s) resident within the platform.
- Note any formal evaluation process that has been used to assess the security and efficacy of the tamper-resistance features of the operating environment – e.g., validation of a TEE implementation through industry best practice testing and/or certification program. It is not a requirement that any such implementation has been assessed through an external approval process, but the assessor must be aware of what testing has been performed to create a valid test process for this requirement. Where previous

evaluation results are to be accepted and/or reused, the tester must validate and provide evidence of their equivalency.

The intent of this test item is for the tester to detail the security features of the PIN CVM application tamper resistance as implemented and relied upon by the operating environment, so that this information may be used during the analysis attempt to break the obfuscation.

TB1.7 The tester shall document any additional protections provided by the application; such protections may include, but may not be limited to, compile-time protection options used, virtualized execution or other hardware level abstraction to the application operation, etc. The tester shall confirm that these protections apply across supported platforms (as assessed under TR B12: Supported Platforms) or detail where any gaps exist in coverage of these protections and justify why this does not increase the risk posed by those platforms.

TB1.8 The tester shall attempt to circumvent the tamper-resistance properties and subvert the normal operation of the PIN CVM application for the purposes of capturing or compromising the PIN entry and processing. This must be done without tamper-detection and response features of the monitor environment active, to demonstrate entirely and solely the protection provided by the tamper-resistance features alone. It is expected that the tester shall consider use of state-of-the-art techniques used in the reverse engineering of malware and attacks on DRM systems and may implement the code outside the COTS device on which it is normally targeted for execution.

The tester shall document the process used and the outcome of this attempt.

TB1.9 Based on the above testing and information, the tester shall provide a costing of an attack to bypass the tamper-resistance properties of the PIN CVM application, using the methodology outlined in Appendix B: Software Tamper-responsive Attack Costing Framework. This costing should not consider the monitor aspect of the solution, only the tamper-resistance features of the local PIN CVM application.

TR B2: Leakage during PIN Entry

The PIN CVM application shall implement methods to mitigate the use of sensors to determine the value of the PIN during entry.

Guidance

Without additional protections, it may be possible for another application executing within the COTS device to recover information about the PIN through monitoring of on-device sensors. The movement of the COTS device during the entry of the PIN can be directly correlated to the location of the customer data entry (and therefore the PIN value). Similar recovery may be performed by using the device camera or microphone to capture the movement of the customer's hands or use of the device camera to capture an image of the customer fingerprint.

The scope of this requirement does not include recovery or capture of the PIN data by mechanisms outside the COTS device or scope of evaluation. For example, capture of the PIN using external cameras, or interference patterns in wireless signal strength at a router, is not in scope.

The tester is required to develop costings for the most feasible attack for this requirement. Although minimum costing point values are not specified in this version of the standard, the quality and efficacy of the monitoring system is considered vital for the security of the overall PIN on COTS solution. Therefore, it is to be considered a non-compliance if values of "low" are assigned to any of the monitor items in the costing produced.

TB2.1 The tester shall document the features implemented by the PIN CVM application to mitigate the use of on-device sensors to determine the value of the PIN during entry.

TB2.2 Where the system vendor claims that the PIN used is not vulnerable to exposure through use of on-device sensors, the tester shall justify and provide test evidence to support whether this is considered to be correct – if so, no further testing is required for this TR. For all systems, PIN entry must meet the provisions of this TR.

TB2.3 Where mitigation features include the use of randomized elements, the tester shall confirm that the random values are obtained from a source that has been assessed to the random data requirements in TR A2: Random Numbers.

TB2.4 The tester shall document all sensors provided by all the supported platforms defined under TR B12: Supported Platforms. This list must not be limited to the example sensors provided in the TR guidance where additional sensors exist in any of the supported platforms.

TB2.5 Where the mitigation features involve the disabling of on-device sensors during PIN entry, the tester shall confirm that this covers all sensors defined in the testing requirement above. The tester shall use this definition of the sensors and vendor mitigation measures to define a test plan to assess the effectiveness of these mitigations. The tester should also consider sensors that the vendor states are disabled or otherwise mitigated and perform tests to assess the efficacy of this mitigation. The scope of this test plan must be justified within this section of the report.

TB2.6 Where the mitigation features involve the disabling of on-device sensors during PIN entry, the tester shall attempt to access sensors that may leak this data during the PIN entry process. If the sensors can be accessed and no further mitigations are applied, this requirement must be marked as non-compliant.

TB2.7 The tester shall confirm that the protections implemented to prevent leakage through the device sensors operate throughout the maximum possible duration of the PIN entry process.

TB2.8 The tester shall detail how the PIN CVM application mitigates the capture of card data through any contactless or NFC interface on the COTS device. This may include (but not be limited to) preventing or detecting enablement of this interface, blocking other applications from accessing the interface, forced transmission of data through this interface, etc. The tester must confirm that the method used is valid both on initial execution of the PIN CVM application and throughout operation of this application.

TB2.9 Based on the above testing and information, the tester shall provide a costing of an attack to use the on-device sensors to extract the customer PIN. This attack must consider the protections provided by all security features, including any tamper-resistance or monitor features which may protect against such attacks. The tester must use the methodology outlined in Appendix B: Software Tamper-responsive Attack Costing Framework for this costing.

TR B3: Protection of Correlatable Data

The PIN CVM application must provide protection to any correlatable data that is stored and processed within the COTS device. The correlatable data must never be exposed or required for operations within the rich execution environment (OS) of the COTS device.

Guidance

To allow for the widest applicability of this standard, an exhaustive list of sensitive data is not provided, and this must be determined by the assessor during the evaluation. However, at all times, this list will include the customer PAN, sensitive account data, any cryptographic keys used or stored by the PIN CVM application (even if encrypted) and the code for the PIN CVM application and attestation component on the COTS device.

The security of COTS-based PIN entry is largely based on the separation of the PIN from the account data, using physical or cryptographic means. Although multiple layers of protection are provided to the PIN itself through compliance to this standard, a fundamental defense against fraud is provided by ensuring that the PIN and account data are kept separate.

This requirement covers the protection of the account data on the COTS device only, and therefore as provided by the PIN CVM application. Protection within the SCRP or back-end monitoring systems is covered under separate TRs. Specifically, customer PIN requirements for the SCRP are covered as an approval class within the PTS POI Security Requirements, and the card data must never be exposed (or available) in plaintext outside this PCI PTS approved SCRP or payment processing environment.

TB3.1 The tester shall confirm that the full PAN and expiry date, as well as track data or track equivalent data, are not available (or required for processing) in clear text nor in encrypted form under any encryption key ever available to the PIN CVM application outside the SCRP.

TB3.2 The tester shall provide a list of all sensitive data considered for this requirement.

TB3.3 The tester shall provide a block diagram that indicates where all sensitive data is available in plaintext on the merchant-side systems. This shall include, but may not be limited to, the Card Reader, the COTS rich execution environment and any TEE or physically separate security processing elements used. This diagram shall indicate the flow of sensitive data through the various elements.

TB3.4 The tester shall confirm that there is no method, function, processing requirement or potential for the correlatable data to be decrypted in some other area of the in-scope systems and returned to the PIN CVM application or COTS rich execution environment in plaintext.

TB3.5 The tester shall confirm that the PIN CVM application never has access to the cardholder data in plaintext, and that any reliance on, processing or authentication of the cardholder data occurs in systems that are protected from being accessed by the PIN CVM application or COTS rich execution environments.

TB3.6 The PIN CVM application must protect against attacks designed to expose data in storage or memory and deploy appropriate controls including minimizing such storage post transaction completion or application time out.

TB3.7 Where the implementation relies upon a TEE, security processor or other security feature of the COTS devices used, the tester shall confirm that all supported platforms as assessed under TR B12: Supported Platforms provide the required features. The tester shall undertake a search of publicly disclosed vulnerabilities for these relied-upon features and confirm that no such vulnerabilities exist that are not suitably mitigated by the tamper-response features (as assessed under TR C2: System Tamper Response).

TB3.8 Where the protection relies upon a TEE, security processor or other security feature of the COTS devices used, the tester shall confirm that these areas do not have access to the customer plaintext card data or any cryptographic keys that may be used to decrypt the account data.

TR B4: Determining Keys Analysis

The PIN CVM application must provide sufficient protection to any cryptographic keys that are used for the protection of the PIN. Cryptographic keys used within the PIN CVM application must be updated at least monthly.

Guidance

This requirement covers the security of the cryptographic keys used for protecting the PIN. This includes not only keys used to directly encrypt the PIN, but also keys used to provide a secure channel to disparate components of the system (such as the SCRP or monitor), or to provide security services to the application itself - e.g., encrypting sections of the code for the purposes of code obfuscation.

This requirement is not intended to cover the security of cryptographic keys in the SCRP or monitor – these items are covered under separate requirements in the approval class for SCRP in PTS POI and TR E2: PCI PIN Compliance.

Where purely software methods are used for the protection of these keys - for example, through use of "white-box" cryptography – the specific instance used in a PIN CVM application must be regularly changed within a period shorter than the expected time required to reverse engineer the software protections. At a minimum, changes to the white-box instance must occur at least once per month. It is acceptable to have two sets of keys during the changeover period but all "old" keys must be invalidated once the new keys are installed. This requirement does not imply that different implementations, or different algorithms, must be used each month.

Use of protection mechanisms that are inherent to the platform on which the PIN CVM application runs (e.g., hardware-backed keystore) may be acceptable in place of white-box cryptography. However, where such device-dependent security is relied upon, testing must be performed on all different types of protection provided. For example, where protection by a TEE is claimed for key security, each platform on which the PIN CVM application runs must be confirmed to provide a secure TEE implementation. Combined approaches that use different protection methods for different platforms (depending on the security features provided by that platform) or hybrid implementations that use combinations of hardware- and software-based protections may also be considered acceptable.

The tester is required to develop costings for the most feasible attack for this requirement. Although minimum costing point values are not specified in this version of the standard, the quality and efficacy of the monitoring system is considered vital for the security of the overall PIN on COTS solution. Therefore, it is to be considered a non-compliance if values of "low" are assigned to any of the monitor items in the costing produced.

TB4.1 The tester shall provide a list of the locations and protection methods provided to all cryptographic keys in the PIN CVM application, where those keys are used for the protection of the PIN or security of the PIN CVM application operation. This must include keys that are used for the security (encryption) of the PIN, as well as keys that are used for establishing secure channels (as per TR D2: Secure Channels), generating or whitening random data and validating authenticity of code and/or datagrams.

TB4.2 For each cryptographic key, the tester shall confirm how often this key value is changed, whether this process is manual or automatic, and that this occurs at least once every month. The method used to update these keys must be detailed in each case. The tester shall confirm that any key rotation utilizes cryptographic keys from the attached SCRP to provide security to the update process.

TB4.3 For each different protection method claimed, the tester shall detail which are provided wholly by the PIN CVM application (such as white-box cryptography) and which rely on protections provided by the platform on which the PIN CVM application is executed.

TB4.4 Where software protections (such as white-box cryptography) are used to protect the cryptographic keys, the tester shall confirm that these code areas/functions are integrated into the overall PIN CVM application and are protected by the tamper-resistance features to prevent easy extraction of the code and/or use of this code as an encryption oracle.

TB4.5 Where software protections (such as white-box cryptography) are used to protect the cryptographic keys, the tester shall obtain and review the full source code for the implementation. The tester shall use this to confirm that the implementation is robust and shall use this analysis as input to the attack and analysis processes required throughout this TR.

TB4.6 For each protection method that relies upon features provided by the platform, the tester shall confirm that all the devices and OS versions of the supported platforms (defined in TR B12: Supported Platforms) provide these features. For each instance, the tester shall note:

- Whether any public vulnerabilities exist for the protection mechanism
- Whether the protection mechanism has undergone any formal certification or validation process
- Which platforms have been explicitly tested by the test laboratory as part of this evaluation process
- What protections are provided against side-channel analysis to recover the cryptographic keys (where the side channel may be timing-, power-, EM-based, etc.)

TB4.7 For each protection method that relies on software-based protections, the test laboratory shall note:

- How the software protection method is implemented, and how this conveys security to the cryptographic keys
- Whether the protection method has undergone any formal certification or evaluation process
- Whether the protection method differs between platforms due to differences in the OSs or other platform-specific features
- How the protection method prevents side-channel and algebraic attacks (such as DCA or BGE). This must include consideration of the use of virtualized environments, monitoring of program address/data access and execution flow using methods such as statistical analysis, code lifting and exploitation of cryptographic oracles/APIs, to recover information about the cryptographic key(s).

TB4.8 The tester shall note any features of the cryptographic implementation that protect against fault injection attacks. It is expected that the tester will consider fault injection through at least (but not limited to) the following:

- Requiring local physical interaction with the device running the PIN CVM application (such as through the use of EM Fault Injection);
- Direct injection of faults into the code execution through manipulation of the execution environment (as made possible through direct control of that execution environment, such as in a virtualized context); or

- Manipulation of the execution environment through an interface to hardware features such as clock/voltage settings, direct/indirect memory access (using "RowHammer-" like attacks), etc.

TB4.9 The tester shall confirm what physical test, debug or in-circuit emulation features exist on the supported platforms and consider these in any attack methods and costings produced.

TB4.10 Based on the above testing and information, the tester shall provide a costing of an attack to determine, extract or modify the cryptographic keys used by the PIN CVM application for security services. The tester shall perform this costing using the methodology outlined in Appendix B: Software Tamper-responsive Attack Costing Framework. This costing should consider the difficulty in bypassing all tamper-resistance and monitor features of the solution, where applicable.

TR B5: Online Processing

The PIN CVM application must only allow for the processing of payment transactions where the financial request message is transmitted to the host online during the transaction. Validation of online connectivity must be performed prior to the commencement of each transaction, and where such a connection is not available; the entry of the PIN must be prohibited.

Guidance

All transactions must be processed online.

In the context of this requirement, "online" refers to the transmission of the financial message to the remote host during the performance of the transaction. Such transactions may utilize either online or offline PIN verification. Where online connectivity is not available, the transaction processing must be prevented. If connectivity drops during a transaction and the transaction must be reversed, all customer data must be erased from the system.

Disabling of software-based PIN entry for magnetic-stripe transactions must be at the level of the PIN CVM application.

TB5.1 The tester shall confirm that the PIN CVM application establishes a secure channel to the back-end monitoring and attestation systems prior to prompting for the PIN entry in each transaction.

TB5.2 The tester shall disable network connectivity on the system and attempt to perform a transaction. For compliance to this requirement, the transaction must not be permitted. Disabling of the network connectively must include at least the following:

- Disabling all network connections
- Deactivating data connectivity for a cellular network
- Connecting to a local network (such as Wi-Fi) that is not connected to the Internet

TB5.3 The tester shall start a transaction where network connectivity is available but disable this connectivity during PIN entry and detail the outcome of the transaction for more than 60 seconds. The PIN CVM application must reverse and/or cancel the transaction and erase all customer data. Any time-out period implemented to wait for reconnection of the network must not exceed 1 minute after PIN entry.

TR B6: PIN CVM Application Authenticity

The PIN CVM application must be installed and updated using methods that ensure its integrity and authenticity, using only the online store provided by the COTS OS vendor implementing cryptographic validation of this OS store and any loaded applications. The system must detect when the PIN CVM application has been side-loaded outside normal channels and treat this as a tamper-detection event.

Guidance

The authenticity of the PIN CVM application is a paramount concern in the security of PIN entry. Loading of applications from the OS store provides a level of confidence that the application has not been tampered with before being installed on the merchant system. The tester must validate that there are no public vulnerabilities for the stores used.

Where the PIN CVM application allows or requires download of additional data from the back-end monitoring systems, such data must also be cryptographically signed and authenticated by the PIN CVM application prior to use or execution. Reliance upon the secure channel that exists between the PIN CVM application and the back-end monitoring systems is not sufficient. This requirement implies that each datagram sent from the device to the back-end server, or vice-versa, has an individual signature or (H)MAC applied. The validation of this datagram signature/MAC must be provided by systems outside the execution environment of the PIN CVM application (e.g., by the external SCRP).

The scope of this requirement is for the authentication of the application by the base OS. Additional authenticity checks are expected to be applied and checked by the monitoring system or applied as part of the obfuscation methods, but these are not in scope of this requirement.

An authenticated mode of encryption that has been approved by NIST or other international standards bodies may be used instead of a discreet signature or (H)MAC.

TB6.1 The tester shall detail all supported methods for loading the PIN CVM application onto the acceptable baseline systems (as assessed under TR C1: COTS System Baseline). This may include multiple online stores, but the tester shall confirm for each instance that only the official store of the baseline OS(s) is used for PIN CVM application deployment.

TB6.2 The tester shall confirm that the online stores for all supported platforms (as assessed under TR B12: Supported Platforms) enforce the use of methods to validate the authenticity and integrity of any connections between a device and the store (e.g., by implementing recent and secure versions of TLS with cipher suites that enforce strong cryptography). This connection must prevent MITM and replay attacks.

The tester shall document the security controls and cryptography enforced by each supported platform in this regard.

TB6.3 The tester shall confirm for each OS store used for PIN CVM application deployment that signature(s) provide for validating the integrity of the application in its entirety. The tester shall confirm that this signature is checked prior to installation of the application and shall make note of any acceptable baseline systems where the signature is not (or cannot be) checked on executable code after installation onto the COTS device (for example, where AoT compile features are implemented, and the resulting compiled code is not provided with cryptographic authentication mechanisms that can be checked on execution). It is not the intent of this requirement to prevent the use of such systems, but it is expected that additional monitor protections may be required in such instances.

TB6.4 The tester shall document any additional scripts, data, executable files, interpreted commands or other information that is downloaded by the PIN CVM application after installation. In each case, the tester shall confirm that data is cryptographically authenticated and the authentication is validated prior to use of the data. The tester shall further confirm that any datagrams sent from the PIN CVM application to the back-end monitoring system are cryptographically signed or MAC'd prior to transmission.

The validation of the additional data and communication datagrams must be performed by the SCRP.

TB6.5 The tester shall confirm that signing of the PIN CVM application must be performed using a dual controlled process, using cryptographic keys that are secured within an HSM formally approved to PCI HSM or FIPS140-2 Level 3 (or above). This requirement does not apply to signatures that can only be generated by a third party, such as the OS store itself. However, all signatures that are generated by the PIN CVM application vendor must meet this requirement (even if used by the OS store).

TB6.6 The tester shall confirm that there are no publicly disclosed vulnerabilities in the OS stores supported that have not been patched or remediated in all implementations covered under the System Baseline. Where vulnerabilities are noted, the tester shall document what, if any, features of the solution mitigate such vulnerabilities.

TR B7: Clearing of Internal Buffers

***Sensitive data** shall not be retained any longer or used more often than strictly necessary. **PIN**s and other customer identifiable data must be encrypted within the **PIN CVM application** immediately after entry is complete and has been signified as such by the **cardholder**, (e.g., via pressing the enter button).*

***The PIN CVM application** must automatically clear the internal buffers it controls when one of the following occurs:*

- ***The transaction terminated for any reason (including success or failure); or***
- ***The PIN CVM application has timed out waiting for the response from the cardholder or merchant; or***
- ***A tamper-detection event has been signaled by the monitoring system, or the PIN CVM application is halted, loses focus or otherwise is moved to background processing.***

Guidance

The vendor shall provide documentation of test results for inspections of internal buffers. It is understood that clearing of buffer data from high-level languages can be complex, but the implementation must perform a rigorous effort to achieve this. Solely relying on "garbage collection" functions for clearing buffer data is not considered compliant.

The device will support the encipherment of the PIN multiple times as part of a transaction series; transferring it between the PIN CVM application and SCRP, then from the SCRP to back-end payment system. Each merchant-side instance must be implemented to clear the buffer as soon as practical after use. Labs must evaluate the methodology used to clear temporary variables, heap, stack and other memory sub-systems of plaintext PIN data to prevent memory dumps from revealing sensitive customer data.

It is expected that lab testing, and vendor evidence will be based on unobfuscated code; although evidence from "production ready" code that has undergone obfuscation compliant with requirement B1 is ideal, it is expected that this will greatly complicate the validation of the buffer clearing functions. The intent of this requirement is to confirm that significant attempts are made at clearing the buffers and use of unobfuscated code is considered to provide the best path to validation.

Where necessary, the vendor shall provide a test harness to facilitate the validation of this requirement. "Witness testing," where the laboratory confirms the clearing of buffers through testing performed by the vendor directly, is acceptable but this must be performed on code running in a physical instantiation of the supported platforms (not a virtualized environment or VM) and the laboratory must specifically note where witness testing was used, why this was necessary, what control/input to the testing the laboratory staff had, that all testing was performed as required in these TRs and how the laboratory staff confirmed this was sufficient.

TB7.1 The tester shall examine and cite any relevant documentation, including vendor test results for inspections of internal buffers, user guides, the software specification or the software implementation submitted by the vendor to verify that it supports the vendor responses.

TB7.2 The tester shall note how each item of sensitive data is managed during the transaction, specifically noting whether any item is located within a non-volatile medium other than the PIN CVM application memory space. Where this occurs, the tester shall confirm that this location is protected through use of encryption. The tester shall note the algorithm and mode of operation implemented and ensure that the cryptographic keys used are detailed under TR A4. The tester shall justify the mode of encryption used does not introduce vulnerabilities that are not otherwise mitigated.

TB7.3 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, plaintext PIN values, plaintext cryptographic keys, customer identifiable data such as e-mail address and phone number, or key components are considered sensitive data.

TB7.4 The tester shall review the source code of the PIN CVM application and confirm that – and summarize how – sensitive information is cleared from all storage locations after use, including local variables (before exiting the function) and other locations. The tester shall detail the methods used to perform the erasure; and where clearing of these memory locations cannot be 100% confirmed, the tester shall document his or her attempts to validate the erasure of sensitive information and provide justification that the methods implemented by the PIN CVM application are sufficient.

TB7.5 The tester shall detail the method used by the vendor to mitigate the potential that this buffer-clearing code/function cannot be removed by compiler optimizations or other means of code optimization, if employed by the vendor. Special note shall be made where the supported platforms include on-device compilation methods.

TB7.6 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 confirm 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

TB7.7 The tester shall review vendor software-development practices documentation to confirm that and indicate how:

- The requirement for buffer clearing is documented including specific notes that buffer clearing should be performed to prevent removal by compiler optimization; and
- The correct implementation of this guide is reviewed as part of the firmware-verification process assessed under TR A5: Secure Software Development Practices.

TB7.8 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.

TB7.9 The tester shall force a tamper-detection event and confirm that this results in the clearing of all customer data from the PIN CVM application. This may require assistance from the PIN CVM application vendor. A summary of the results is to be provided with details on which areas rely on vendor testing, and which have been directly tested by the assessor. The tester must document the testing and validation process, providing evidence of the erasure of sensitive data.

TR B8: Secure Provisioning

The system must allow for the secure provisioning and initialization of the PIN CVM application on the COTS system after download. This will include the deployment of initial device-unique cryptographic keys and other merchant-unique data.

Guidance

As the PIN CVM application is downloaded from a single instance in an OS store, each application install is initially identical to all others. The system must implement methods to immediately upon install ensure that instance is unlike any other and can be uniquely identified and secured through communication with the back-end monitoring and attestation systems.

TB8.1 The tester shall outline the methods implemented to securely provision cryptographic keys and other data to the PIN CVM application upon initial configuration. This must include the use of cryptographic keys stored within the SCRP to provide security to the provisioning process.

TB8.2 The tester shall detail the provisioning process using a message flow diagram and confirm that this occurs over a secure channel that protects against MITM and replay attacks, as well as protecting the confidentiality and integrity of the data communicated. Where a standard protocol (such as TLS) is not used, the tester shall note which areas of the messages provide the MITM and replay protections. The tester shall confirm that only approved cryptography is used, as per Appendix C: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms.

TB8.3 The tester shall install the PIN CVM application and monitor the provisioning process (e.g., through a network monitor observing traffic from the device) and confirm that a secure channel is used to prevent oversight or manipulation of the provisioning process.

TB8.4 Where white-box cryptography is used, the tester shall outline how cryptographic keys that are unique per application installation instance are conveyed and secured using this method, such that use of the common white-box keys is minimized after the secure provisioning process. The details provided must include how each white-box key is used, how they are involved in the secure provisioning process and what use the white-box keys have (if any) after the secure provisioning process is completed.

TB8.5 Where white-box cryptography is used, the tester shall detail the process used for updating the white-box cryptographic keys in the application. The tester shall detail how any method used for ensuring that current working keys are not erased or rendered inoperative during this process provides for the confidentiality and integrity of these keys.

TB8.6 The tester shall confirm that no secret or sensitive data is transferred prior to the secure establishment of cryptographic keys unique to each PIN CVM application installation, and that any key-generation or key-agreement methods used in the secure provisioning are implemented securely and have been assessed against the key-management requirements of this standard.

TB8.7 The tester shall confirm that the secure provisioning process implements principles of perfect forward secrecy to ensure that any future compromise of the initial keys used during the provisioning process does not expose keys that have already been established and/or used.

TR B9: Separation of PIN Processing Code

Software that handles, secures or otherwise affects the security of the PIN entry and processing on the merchant device must be logically separated from code that is used for other purposes (such as general merchant UI).

Guidance

Areas of the PIN CVM application that are not directly involved in the processing or handling of the PIN or providing security services and interface to the monitoring system must be able to be updated without affecting the PIN CVM application secure code. This isolation may be achieved in many ways, but simply having different functions within the same body of code for security and non-security functions is not considered sufficient isolation.

Multiple layers/levels of separation of the code may be implemented – for example, the code used for cryptographic key storage may be logically isolated from code used for PIN data entry, and both of these code segments logically isolated from the overall merchant UI code.

TB9.1 The tester shall detail how the code of the PIN CVM application is structured and confirm that methods are provided to logically separate the PIN processing code (or code that provided security to the PIN processing code) from other parts of the code. This must include data-flow diagrams that show how the PIN is entered, processed, encrypted and validated within the PIN CVM application, where the data is transmitted outside of the scope of the PIN CVM application and any assumptions made about these external connections.

TB9.2 The tester shall justify how this separation allows for easy isolation of code segments, and therefore ensures that changes made to one area of the code do not affect areas that are isolated from this. Where the testers cannot provide this justification, or where code isolation is provided through simple use of different functions or code files, this requirement must be marked as non-compliant.

TR B10: Secure PIN Entry Process

PINs must be entered in a manner that protects their confidentiality and integrity.

Guidance

The PIN entry process must be protected against manipulation or subversion. Attempts to modify, replace or subvert the customer prompts, keyboard or other UI features that are important for the security of the system must be prevented.

Where secure data is entered into the system, such as customer PINs, the "keyboard" or standard data entry functions of the OS must not be used (i.e., touch location data alone must be collected and then translated into PIN values within the PIN CVM application).

This requirement is to be assessed against all OS types and variants that exist under the supported platforms (as assessed under TR B12: Supported Platforms). The scope of this assessment should not consider malicious modification of these supported platforms to enable such features specifically for the collection of customer PIN data. However, modifications to a "standard" OS that may be performed by a COTS vendor, distributor or carrier that are not necessarily specifically designed for PIN capture, but which may be maliciously repurposed or used for such, are to be considered in scope.

The intent of this requirement is to validate that existing functions or APIs that can be maliciously repurposed by another resident application do not exist within the supported platforms.

PIN entry must only be allowed for an EMV-based contact or contactless transaction.

The tester is required to develop costings for the most feasible attack for this requirement. Although minimum costing point values are not specified in this version of the standard, the quality and efficacy of the monitoring system is considered vital for the security of the overall PIN on COTS solution. Therefore, it is to be considered a non-compliance if values of "low" are assigned to any of the monitor items in the costing produced.

TB10.1 For each supported platform (as assessed under TR B12: Supported Platforms), the assessor shall note what protections are provided to prevent another application from:

- Stealing focus during PIN entry, to capture the customer data as it is entered
- Stealing focus at any other time during the foreground execution of the PIN CVM application to maliciously prompt for (and thereby capture) PIN entry
- Screen capture during PIN entry

TB10.2 The tester shall confirm that the PIN CVM application does not allow for the entry of the PIN unless there has been a non-magnetic-stripe card read through the attached SCRP for that transaction. The tester shall note any other transaction types supported by the PIN CVM application, and for each type that does not involve a fully compliant EMV-based transaction, confirm that PIN entry is not supported or able to be performed.

TB10.3 The tester shall confirm through testing of representative samples of the supported platforms that switching context from the PIN CVM application to another application during PIN entry results in the termination of the PIN entry process and deletion of any partially entered PIN data. Tests shall include both manually changing focus (e.g., switching to another application) and automatic changes of focus (e.g., during an incoming phone call or text message).

TB10.4 Where system messages provide pop-up, pop-in or pull-in dialogs that cannot be detected or disabled, the tester shall justify why these notifications cannot be used to steal the PIN data as it is entered (by displaying a fake keypad in an attempt to have the customer mistakenly enter their PIN data into the fake keypad rather than the secured PIN CVM application keypad).

TB10.5 Where the baseline systems support windowed environments, where multiple applications can be displayed on the screen at any one time, the tester shall confirm that the PIN CVM application does not allow for PIN entry unless it is running in full-screen mode.

TB10.6 The tester shall confirm that:

- The PIN CVM application does not use the "standard" OS keyboard or other entry methods that directly return alphanumeric data that reveals the PIN provided (i.e., by collecting only touch location data that is then processed into customer PIN values)
- The data is not passed or otherwise made available/accessible to any other application
- The values entered by the customer are not displayed on the merchant screen (even temporarily). Only non-deterministic values, such as an asterisk, may be displayed.
- Any tones provided during the PIN entry are not identifiable to the value entered (e.g., not DTMF tones or other sounds that can be correlated to the (virtual) button pressed by the customer)
- No part of the PIN is transmitted outside the COTS system prior to the entry being completed by the customer and the entire PIN encrypted as per the requirements of this standard. Touch locations, individual data elements, images, etc. of the PIN must not be transmitted externally to the COTS device unless these constitute the entirety of the PIN entry.
- Localized indications of customer interaction (such as button "highlights" when pressing an area of the screen, localized haptic feedback, etc.) must not be used.

TB10.7 The tester shall attempt to capture the screen during PIN entry, using both native OS methods and additional applications. Where screen capture is found possible, the tester shall detail how this is detected and/or mitigated by other security aspects of the solution and ensure that consideration of this is included in the required costing performed to validate compliance to this TR.

TB10.8 Based on the above testing and information, the tester shall provide a costing of an attack to determine the value of the customer PIN. This attack must consider the protections provided by all security features, including any tamper-resistance or monitor features that may protect against such attacks. The tester must use the methodology outlined in Appendix B: Software Tamper-responsive Attack Costing Framework for this costing.

TR B11: Sanitization of Customer Data

Customer data must not be stored, logged or otherwise maintained within the PIN CVM application. Where such data is sent from the back-end monitoring system for display or printing in the merchant environment, it must be truncated to ensure that it cannot be uniquely correlated with the customer and the full details are not available to the merchant. Entry of customer identifiable data on the merchant system must only be required for the supply of receipt data and must be deleted from the merchant system immediately upon completion of the transaction.

Guidance

It is a fundamental security tenet of this standard that the PIN data remains logically isolated from the account data. However, collection of account data may occur in other systems or locations, and then be correlated with the PIN later using information about the customer, such as name or e-mail address. Therefore, it is important that all identifiable customer data be prevented from being exposed on the COTS system.

The customer PAN may be provided to a maximum of the last four digits, and customer phone numbers must display, at a maximum, only the last 4 digits, and e-mail addresses must display, at a maximum, only the first two characters of the address and first two and last two of the domain (e.g., anxxxxxxxxxxx@gmxxx.xom or jkxxxxx@soxxxxxxx.xxx.au). Additional privacy controls may be required by regional/country regulations.

Truncation must not occur in the COTS rich execution environment.

TB11.1 The tester shall confirm what customer data is accepted and used by the system, either in the merchant environment or on any back-end server systems. For each item of customer data, the tester shall identify how this data is provided.

TB11.2 Where customer data (such as e-mail address) can be provided into the PIN CVM application, the tester shall confirm how this data is erased after being transmitted from the application, such that it is irrecoverable on the merchant systems.

TB11.3 The tester shall confirm that any transmission of customer identifiable data occurs across a secure channel as tested under TR D2: Secure Channels.

TB11.4 The tester shall perform payment transactions and confirm that customer data is either not displayed or presented in a truncated format. The tester shall confirm that this truncation does not occur on the merchant systems but at the back end, such that full customer data is not exposed on merchant systems.

TB11.5 The tester shall review any logs produced by the system and confirm that they do not store un-truncated customer data.

TB11.6 Where customer data is entered into the PIN CVM application, the tester shall confirm that the entry methods implemented do not allow for the caching or storage of the customer data into the COTS OS (e.g., dictionaries, contact database, OS "clipboards," etc.).

TR B12: Supported Platforms

There must be a clear definition of all platforms – including device types, hardware and OSs – on which the PIN CVM application may be executed, and for which all other PIN CVM application security controls (as assessed under the "B" section requirements) are valid.

Guidance

Although these requirements are designed to allow for the entry of a PIN on a COTS platform that has not been directly assessed for security, it is expected that the system will have criteria for which platforms are considered acceptable for PIN use, and which are not. For example, it is expected that systems using older COTS OSs that may contain unpatched vulnerabilities will not be deemed acceptable for PIN entry. The PIN CVM application vendor must provide a clear methodology for determining the suitability of any COTS platform; this may be a whitelist, blacklist or hybrid approach, but must clearly demonstrate a risk analysis of the platform that accommodates any known or potential vulnerabilities in each merchant device.

There are two requirements in this standard that address the suitability of devices/OSs on which the PIN CVM application may be executed. This "supported platforms" requirement addresses the need for the PIN CVM application to be targeted to a limited subset of all available devices, and that the PIN CVM application developer must have undertaken some risk analysis and mitigation steps to identify which platforms are suitable and secure.

The "system baseline" requirement (TR C1: COTS System Baseline) addresses the need for the monitoring system to dynamically assess the security of all devices/platforms on which the PIN CVM applications are executed and determine whether they are vulnerable to attack. It is expected that the system baseline will change more rapidly than the supported platforms to address the changing threat landscape.

TB12.1 The PIN CVM application vendor must maintain a defined list of supported platforms. The tester shall confirm that this list exists and detail the methodology for determining whether a platform is acceptable or not. The tester shall specifically identify what security process is involved in the identification of supported platforms, and the method used for platform selection (whitelist, blacklist, etc.).

TB12.2 The supported platforms must only include products and OSs that provide the following features:

- An enforcing mandatory access control framework
- A "trusted boot" mechanism that validates the OS authenticity from a hardware root of trust. This mechanism must either prevent the loading of unauthentic OS code or allow for detection of such code by the monitoring system.
- Ability to prevent all other applications other than the foreground application from accessing touch-event details
- Validation of an application signature upon loading and execution of that application. This signature must be calculated with strong cryptography and no known bypass measures may exist for the acceptable baseline systems.
- Isolation of touch-event data such that only the application currently in focus can receive or otherwise identify the location of touch events
- Full screen operation of applications

TB12.3 The tester shall confirm that validation of the supported platform is always performed:

- When the PIN CVM application is executed
- As required by the monitoring system
- Whenever white-box cryptography or obfuscation methods, implementations or instantiations are updated

TB12.4 The tester shall attempt to install the PIN CVM application and perform PIN entry on a COTS device that is not within the supported platforms. This device should be selected to be a "close fail" – that is, a device that is common in the market but "just" falls outside of the acceptability criteria (or a device that was initially within the acceptability criteria but has been modified to fall outside these). If the install is successful and transactions can be performed, the requirement should be failed.

TB12.5 The vendor shall provide to the lab a document that indicates all security features of the supported platforms that are relied upon for the secure operation of the PIN CVM application, including variances in any minor OS versions or device manufacturer implementations. The tester shall validate that this list is complete and that all such security features are present, operational and provide the required security on the supported platforms.

TR B13: Security of PIN Processing and Transport

The PIN must be secured by SCDs or strong cryptography at all points after being sent from the COTS device to the SCRP.

Guidance

Customer PINs must be encrypted using PIN blocks approved under ISO 9564. ISO format 4 must be used when transferring customer PINs between the PIN CVM application and SCRP. ISO formats 0, 3 or 4 may be used when transferring the customer PIN from the SCRP to the back-end payment system. ISO Format 1 is forbidden from use.

ISO Format 2 may only be used to transfer the customer PIN from the SCRP to the customer ICC. ISO format 2 must not be used to convey customer PINs from the SCRP to the back-end payment system, or between this system and the SCRP.

For systems that use a "split" or partially cloud-based kernel, details must be provided on where the customer PIN is handled at all points, and systems where the customer PIN is transferred from the merchant device to the remote kernel for processing must use ISO format 4 for transfer of the customer PIN.

TB13.1 The tester shall confirm that the customer PIN is encrypted using ISO format 4 for transport between the PIN CVM application and the SCRP, where it must be translated into ISO format 0, 3 or 4. The use of TDES for encryption of the customer PIN on the SCRP is the only exception allowing the use of TDES in this standard.

TB13.2 For PIN blocks generated within the PIN CVM application, the tester shall confirm that only ISO format 4 may be used, and that the PAN value is supplied by the SCRP as a secure PAN token, using the SCRP PAN token generation feature.

TR B14: Security Policy

The vendor shall create and maintain a security policy that details the acceptable uses of the PIN CVM application, as well as any obligations and procedures the merchant must undertake to ensure security of the PIN entry.

Guidance

The vendor must supply documentation to allow the *merchant to understand the context of the approval of the PIN CVM application, to ensure that it does not deploy or use the system in a non-compliant way. In this context, the security policy informs the merchant both how to use and maintain the system securely and also of any assumptions made by the laboratory during the PIN CVM solution evaluation (e.g., that the device is handheld).*

TB14.1 The tester shall confirm that a security policy exists for the system, and that this is provided to all users of the system and required to be read and accepted as part of the PIN CVM application EULA.

TB14.2 The tester shall confirm that the security policy provides details on all additional components required for the use of the PIN CVM application, including all SCRPs approved for use. A picture must be provided of each SCRP, along with any guidance required for the secure use of that SCRP. Each SCRP must be identified by model, hardware version and firmware version to allow for unique validation of each device on the PCI SSC PTS approval listing.

TB14.3 The tester shall confirm that the security policy explicitly notes how the implementation ensures privacy of the customer PIN entry process. Where the security policy does not specifically forbid the COTS device from being affixed to a stand or counter, the policy must outline how the device must be installed and used to ensure privacy of the PIN entry process.

TB14.4 The tester shall confirm that the security policy informs the merchant that it is responsible for the security of the installed environment and use of the PIN CVM application and COTS device on which it is executed.

TB14.5 The tester shall verify that the security policy includes information indicating that the merchant may be disconnected when vulnerabilities are found on their COTS platform that could result in the exposure of PIN data.

TR Module 3: Back-end Systems: Monitoring/Attestation Requirements

TR C1: COTS System Baseline

A defined set of COTS devices and OSs must exist on which the PIN CVM application may be installed and executed. This system baseline must be validated by the monitoring system on first install of the PIN CVM application. A documented process must exist for updating the system baseline as new threats are identified.

Guidance

The system baseline is a subset of the supported platforms (assessed against the PIN CVM application under TR B12: Supported Platforms). The monitoring system is required to define which of those supported platforms may be secure for use by the PIN CVM application, based on data about current attack methods, new vulnerabilities, etc.

It is expected that this system baseline will change over time, and so the process performed by the monitor to determine the system baseline will not be a single "point in time," but instead an ongoing process that continually assesses the threat environment and allows for decisions to be made about the suitability of PIN entry as an ongoing concern.

TC1.1 The tester shall detail the processes implemented by the system vendor to determine the system baseline for its acceptance of COTS devices. The tester shall detail whether this takes a whitelist, blacklist, hybrid or other approach, and how this accommodates known and potential vulnerabilities in systems.

TC1.2 The tester shall outline which aspects of the baseline validation process are performed on the COTS app itself, and which are performed by other systems or execution environments.

TC1.3 The tester shall verify that validation of the system baseline is performed regularly – at a minimum, during each of the back-end monitoring system checks – and that it is always performed upon the execution of the PIN CVM application.

TC1.4 The tester shall attempt to install the PIN CVM application and perform PIN entry on a COTS device that is not within the baseline acceptability criteria. This device should be selected to be a "close fail" – that is, a device that is common in the market and listed in the supported platforms (as assessed under TR B12: Supported Platforms) but "just" falls outside of the acceptability criteria (or a device that was initially within the acceptability criteria, but has been modified to fall outside these). The tester shall detail the process and devices used for this test.

TC1.5 Where the system baseline accepts the use of COTS products that are no longer supported with security patches by the product vendor, justifications shall be provided for why the acceptance and use of such platforms for accepting PIN entry does not increase the risk of PIN exposure or subversion of the payment process, beyond use of devices which are supported by security patches.

TC1.6 The tester shall confirm that the system baseline does not include devices that are rooted or jailbroken. The tester shall outline what protections are provided to detect rooted or jailbroken environments. The tester shall detail how effective this is expected to be and perform tests to attempt to bypass the detections and note the results.

TC1.7 The tester shall attempt to subvert and bypass the baseline detections by installing and operating the PIN CVM application on platform(s) that are outside the baseline acceptability criteria but have been modified in an attempt to appear acceptable. The tester shall detail the process performed and outcomes of this test.

TC1.8 The tester will confirm that documented procedures exist, and are demonstrably in use, for how changes to the system baseline are managed. For example, it is expected that such changes will result in the effective "disconnection" of some merchants using systems that may have previously been acceptable, but which may now fall outside of the acceptable system baseline. The vendor must be able to demonstrate a process for managing such instances (as well as other events that may result from changes to the acceptable baseline). Procedures that rely on waiting for potentially vulnerable systems to become less common and unused by merchants are not to be considered compliant to this requirement.

TR C2: System Tamper Response

The monitoring system must implement methods to actively detect and respond to events that indicate that the COTS device is being, or has been, maliciously altered. This tamper-detection and response method cannot be wholly instantiated within the same execution environment of the PIN CVM application. The SCRP must disable operation without regular validation from the monitoring system.

Guidance

The system must maintain processes to detect and respond to attempts to modify the PIN entry functions or the security thereof. This response must ensure that further attempts to modify or analyze the code execution are prevented; for example, by preventing any further connections or acceptance of data from the device by the host system or sending a "disable" command that erases sensitive data such as cryptographic keys from the application and prevents further of PIN entry.

Tamper response should prevent simple bypass methods such as clearing the application data, re-installing the application or changing merchant details such as (e-mail) address and name. Where multiple devices may be utilized as part of an attempt to attack the system, the tester shall outline what protections are provided to complicate the use of knowledge gained from one device set (COTS device and reader) in furthering an attack on another device set.

Tamper detection shall not rely on any single feature or response from the operational environment, but instead must be based on checking of multiple features. Where possible these features must include intrinsic, hard-to-copy features of the operational environment (e.g., by using hardware-based information such as processor/GPU/memory speed, thermal response under load, memory utilization, etc.).

The system tamper response must be an evolving process that is designed to detect and adapt to new threats. This process of adaptation must specifically address threats to the PIN CVM application and the PIN, and although this requirement does not prevent the use of "generic" platform attestation methods that may be implemented to detect the presence of malware and Potentially Unwanted Programs, it is expected that such methods will not be sufficient in and of themselves to meet the requirements of this TR, as they may not be specifically identifying all items that may relate to attempts at bypassing the controls required in software-based PIN entry COTS solutions.

As part of this tamper detection and response, it is a requirement that the application is only ever able to function and accept PIN entry when it can connect and authenticate to the remote server. Each PIN-based transaction must result in full communication with the remote host, either as a finalized transaction or as a transaction where a reversal process has been initiated and completed due to communication errors. This ensures that each transaction can be monitored and validated by the system tamper response before allowing any additional PIN transactions to be processed.

TC2.1 The tester shall detail the processes implemented by the system vendor to detect and respond to attempts to tamper with the system. The tester shall note where each control is implemented (in the app, in the monitor, remotely or locally to the merchant device, etc.).

TC2.2 A documented system tamper-response policy must exist that defines health-check rules for SCRP, COTS device platform and monitor mechanisms. This policy must include detailed response procedures for successful or non-successful results. The policy must be maintained and communicated to applicable personnel.

TC2.3 Tamper-detection and response features must be implemented to occur regularly, and in accordance with the documented policy. At a minimum, the tamper detection must occur:

- At initialization/initial installation of the PIN CVM application
- Whenever the SCRP is physically or logically disconnected or re-connected to the PIN CVM application
- On execution of the PIN CVM application, to be completed at least immediately prior to the entry of the first PIN
- If initiated by the monitoring system, back-end server or other component of the overall solution (including the PIN CVM application itself)
- Randomly through PIN CVM application operation, but no less than every 5 minutes of operation (if not otherwise activated by one of the events above during that time)
- Any time the PIN CVM application either loses focus or regains focus
- After changes have been made to any component of the solution

TC2.4 The tamper-response mechanisms must be able to detect and determine the validity of the SCRP being used. At a minimum, this must confirm:

- That a valid SCRP is being used (e.g., the device model and hardware version are correct), this device has been approved for use with the COTS payment acceptance system and the correct firmware version is being used
- That the SCRP is in a secure, operational state, and card data encryption (SRED) functions are active to ensure the encryption of all payment card data output from the SCRP
- That pairing as detailed in TR D1: Pairing of Disparate Components is in place and the correct unique identifiers of the SCRP (such as unique serial number) are present

TC2.5 The monitoring system/system tamper response must be able to identify the merchant involved in the transaction, and detect anomalies using at least (but not necessarily only) the following information:

- Merchant ID and
- Merchant name/business name and
- Merchant physical and logical location identifiers (e.g., GPS location, IP address)

The tester shall detail what information is used for this merchant identification and determine whether there are any publicly available methods to bypass or subvert these identifiers.

TC2.6 The tester shall outline what details, and configuration/operational information, are provided to the tamper-response system about the COTS device being used. Justification shall be provided for why this is considered sufficient, or insufficient, to detect malicious tampering of the operational environment of the COTS device, including detection of emulated and virtual environments. The monitor must collect device-specific information that allows for unique identification of the system. It is expected that this may include (but not be limited to):

- Hardware-based information such as processor/GPU/memory speed, thermal response under load, memory utilization, etc.
- Software-based information such as memory layout/mapping features, process linking, versions/fingerprints of software libraries and OS modules, etc.

TC2.7 The tamper-detection system shall provide features to detect and respond to at least the following:

- Modification of or to the COTS OS such that it deviates from the supported platforms defined in TR B12: Supported Platforms
- Operation of the PIN CVM application whilst the COTS system is in developer, debug, accessibility or other privileged modes that may lead to leakage of sensitive information
- Execution of the PIN CVM application within an emulated, virtualized or modified environment
- Use of "hooking" or other data interception frameworks
- Modification, tamper or other methods for altering the normal execution flow of the PIN CVM application (including unexpected delays or errors in processing)
- Use of the PIN CVM application code, or part thereof, within another (valid and/or invalid) operational environment (e.g., through "code lifting" of the entire, or partial, PIN CVM application to another COTS system after initialization and personalization of the PIN CVM application for a specific merchant)
- Changes to system configuration (changing of COTS device, SCRP, merchant ID, etc.)

The tester shall attempt to execute the PIN CVM application in each of the modified environments noted above and confirm that the tamper-response mechanisms correctly identify the changed operational environment and prevent PIN entry.

TC2.8 The tester shall confirm that the any tamper-detection code implemented in the PIN CVM application is protected by the tamper-resistance features assessed under TR B1: PIN CVM Application Tamper Resistance.

TC2.9 The tester shall note whether it is possible to perform "relay" attacks on the PIN entry, by performing or enabling remote screen display and data input capture of the PIN CVM application on another device (which is therefore not running the system tamper-response code) or another application. The intent of this requirement is to confirm whether this is possible on any of the supported platforms, and if so, the tester must detail what features of the monitor or PIN CVM application prevent the exploitation of such features.

TC2.10 The tester shall note whether changes to the operational environment of the PIN CVM application are detected if made in between checks performed by the tamper-response system. For example, the tester shall enable a high-privileged mode (such as "root") on the system when the PIN CVM application is not executing and revert these changes prior to execution of the PIN CVM application. Where such changes are not detected, the tester shall justify why (or why not) the security of the application can be maintained despite these periods of no tamper detection.

TC2.11 The tester shall attempt to make modifications to the execution environment of the PIN CVM application to recover sensitive information, and document where these attempts are detected by the system tamper response and the outcome of this detection.

TC2.12 The tester shall attempt to install the PIN CVM application on another device of the same model and version of that used to initially install and provision the application. It may be necessary to receive specific assistance from the vendor to facilitate this testing, to disable any non-monitor features that prevent extraction of the application and installation on a different device. The tester shall note whether the application and/or monitor is able to use its unique identification features to detect that it is has been installed on a physically different device.

TC2.13 Where the system tamper response involves a manual process (e.g., a potential tamper event is escalated to vendor staff to validate), the tester shall confirm:

- That written procedures for these events exist and are demonstrably in use. These procedures must cover events where staff members relied upon for such determinations are unavailable.
- Events must be escalated for manual review immediately. The maximum time that any such event can go un-actioned does not exceed 48 hours. Automated systems must be in place to disable any further payment processing from systems when this happens.
- Staff members involved in the tamper-response functions are trained and knowledgeable in the security of all systems covered by the system baseline (as defined under TR C1: COTS System Baseline. Evidence of staff competence must be provided – e.g., through documented output of staff testing (not just training) performed at least annually, acceptance of senior member of staff to speak at well-known security conferences (about security of the platforms used), etc.

TC2.14 The tester shall review the terms of use of each platform and OS store used by the PIN CVM application and confirm that the methods used for tamper detection do not violate this requirement. Where an apparent conflict is found, the tester shall confirm that this does not require that the security feature be removed from the PIN CVM application. The intent of this requirement is not to check that the PIN CVM application meets all relevant legal requirements, but that specific OS store requirements do not mean that security features must be disabled or abandoned.

TC2.15 The tester shall confirm that the system monitor performs an integrity and version check on the PIN CVM application as part of the tamper-detection process. integrity checks must implement only approved algorithms, as per Appendix C: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms.

TC2.16 The tester shall attempt to perform a PIN transaction using a previous version of the PIN CVM application and confirm that the PIN entry process is not permitted until the PIN CVM application is updated.

TC2.17 The tester shall confirm that the PIN CVM application version identifier is protected by the tamper-resistance features of the application. The tester shall attempt to change the version identifier of a previous version of the PIN CVM application and confirm that this does not allow for the execution of the PIN entry process.

TC2.18 The tester shall perform transactions during the time that the monitoring system is being executed and confirm that the monitor process does not interrupt the transaction processing.

TC2.19 The tester shall confirm that each successful attestation process by the monitor ensures that the SCRP is enabled for operation for no more than twice the maximum monitoring frequency (where the maximum monitoring frequency is 5 minutes). The tester shall document the

methods implemented to ensure this and confirm how this implementation prevents replay, pre-calculation or other attacks to bypass this control. The tester shall confirm that any disablement of the SCRP is not solely managed by the COTS device such that even in cases where the COTS platform is entirely compromised, a disabled SCRP cannot be re-enabled to accept card transactions. The tester shall attempt to bypass the methods implemented and document his or her findings.

TC2.20 The tester shall document the features of the monitor token that prevent replay of this value. The tester shall attempt to replay the monitor token and confirm that this does not allow for the continued operation, or re-enablement, of the SCRP.

TR C3: Integrity of Monitoring Systems

The integrity of the monitoring system must be maintained at all times. Manual changes to the system must follow a documented process that requires authorization of two individuals. Where automated changes are permitted, such as through a mechanism that employs machine learning, measures must be in place to prevent the exploitation of this update mechanism.

Guidance

The integrity of the COTS system relies upon the ability of the monitoring system to detect malicious changes or modifications to that system beyond the acceptable baseline criteria. Any changes to the system must be managed through a dual-controlled process to ensure that a malicious insider is not capable of subverting this system.

Where machine learning or other methods are employed to allow the monitoring system to automatically adapt to changes in the risk landscape, protections must be put in place to prevent "data poisoning" or other types of adversarial manipulation of input data to cause invalid rules to be put in place by the system.

TC3.1 The tester shall document how the monitoring system is instantiated and how updates are performed. Updates may involve both manual and automated processes.

TC3.2 The tester shall confirm that file integrity monitoring is used to protect configuration files, executables and public keys/certificates used for security services on any back-end components of the monitoring/attestation system.

TC3.3 The tester shall confirm that, for any manual updates of the monitoring system, there is a documented procedure and that deployment of changes to the production environment requires dual control or equivalent methods.

TC3.4 For any automated methods of update used on the monitoring system, such as machine learning processes, the tester shall detail what protections are provided to prevent attacks that attempt to exploit this automated system through data poisoning attacks (where deliberately invalid data is supplied into the system to bias the detection algorithms incorrectly).

TC3.5 The tester shall confirm the monitor operations staff are adequately trained on the administration and operation of back-end monitoring system.

TC3.6 The tester shall confirm that any back-end components of the monitoring system are maintained in an environment compliant to the requirements of Appendix A: Monitoring Environment Basic Protections. The tester shall review the assessor reports output of this section and confirm that it covers all parts of the monitoring system, that the audit has been performed by a PCI SSC-qualified assessor and that the audit report was based on work performed, and signed by this assessor, no more than 12 months ago.

TC3.7 The tester shall confirm that any parts of the monitoring system that exist on the COTS device are protected by the tamper-resistance properties tested under TR B1: PIN CVM Application Tamper Resistance, or shall perform a separate evaluation of the monitor code against these requirements.

TC3.8 The tester shall confirm what mechanisms exist for full traceability of changes and for the system to be reverted to a known-good state in the case of failure, malicious manipulation or other corruption of its correct operation.

TC3.9 The tester shall verify a mechanism exists for producing audit logs.

TR C4: Risk Analysis and Update Policy

The vendor must have a documented risk-assessment policy and procedure, which is reviewed at least annually. This policy must include the methods used to assess ongoing risk to the solution, how and when updates to the system baseline are performed and how such changes are communicated to affected merchants.

Guidance

As the security landscape changes, platforms or OSs that may be acceptable under the system baseline may become vulnerable. A documented policy and procedure for assessing these changes must exist and must provide details on how decisions are made to remove previously acceptable platforms from the system baseline. Such changes will affect merchants using these platforms, so the documentation must also include how merchant communication is handled in these cases.

It is not considered acceptable for the vendor policy to require a minimum number of merchants to be using a vulnerable platform before it is removed from the system baseline.

TC4.1 The tester shall obtain and review the vendor's risk-assessment policy and update procedure documents and confirm that they contain information on:

- How to assess whether newly exposed vulnerabilities pose a risk to platforms
- The need to reassess all supported platforms at least every year and the method used for reassessment
- How and when updates to the system baseline are performed

TC4.2 Where possible, the tester shall compare the information in the policy with actual changes made to the system baseline to confirm that the policy is being correctly followed.

TC4.3 The tester shall confirm how merchants are informed when changes to the system baseline affect their systems.

TR Module 4: Solution Integration Requirements

TR D1: Pairing of Disparate Components

The PIN CVM application and SCRP must be uniquely identified and paired, and this pairing must be validated by the back-end monitoring/attestation system at each startup of the application, as well as during any tamper-detection scans. A secure channel must be established between each component of the system to protect the confidentiality and integrity of data communications between these components.

Guidance

It can be expected that attacks against software-based PIN CVM solutions include the attempt to bypass security in the SCRP. Such attempts may not be initially successful, and therefore changing between different components by a single merchant, or changing of the same component between different merchants, may be an indication of attempted system compromise. It is not a requirement that detecting changing of these devices or merchants automatically results in the blocking of that merchant account, but documented procedures must exist for escalation of such events to determine whether further action is necessary.

The connection between system components must be secured through use of cryptography. This connection must be implemented to prevent MITM and replay attacks on data, as well as preventing traffic analysis to determine the type or content of data communicated between the components.

TD1.1 The tester shall confirm that mechanisms exist to uniquely identify and validate the SCRP and PIN CVM application as authorized. The tester shall detail these mechanisms, as well as the criteria used to authorize their use.

TD1.2 The tester shall confirm that the system implements and enforces methods for the PIN CVM application to validate the SCRP prior to communication of any sensitive data with that SCRP, or performance of any payment transactions.

TD1.3 The monitoring system must be able to associate the PIN transaction to a specific merchant, COTS and SCRP combination for tracking. If not successful, the transaction must fail. The tester shall document the methods implemented for this identification, as well as attempts made during the evaluation to impersonate the merchant on a different system or otherwise bypass the identification methods.

TD1.4 The tester shall note what methods are implemented to prevent traffic analysis for the purposes of determining sensitive information on each of the paired interfaces present in the solution. It is expected that this will require monitoring of the interface during operation.

TR D2: Secure Channels

Secure channels must be established to protect all communications between the PIN CVM application, SCRP, back-end processing systems and back-end monitoring systems. These secure channels must be implemented to prevent MITM and replay attacks and provide mutual authentication and unique identification of each component.

Guidance

A secure channel is a communications connection between two points that is secured using cryptographic techniques. Separate secure channels, using unique cryptographic keys, must be established between the SCRP and PIN CVM application, as well as the PIN CVM application and monitor, and PIN CVM application and payment network.

The secure channels must provide mutual authentication as well as the ability to uniquely identify each component, so that changes of components can be detected and potentially flagged as tamper events by the monitoring system.

Where standard protocols provide secure channels, such as TLS, certificate pinning (or other applicable methods) to ensure that only authorized connections are possible, such protocols must be implemented. Cryptographic methods must be limited to support only acceptable cryptography as defined under this standard.

TD2.1 The tester shall outline logical connections between the PIN CVM application and other components of the system, using diagrams where appropriate. For each connection, the tester shall detail the method used to provide a secure channel between the PIN CVM application and that component.

TD2.2 For each secure channel implementation, the tester shall confirm how data confidentiality, integrity and authenticity are maintained. The tester shall confirm that all cryptographic algorithms used for these purposes meet the requirements of Appendix C: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms.

TD2.3 For each secure channel implementation, the tester shall verify that the protocol used provides for mutual authentication of the two components prior to communicating any sensitive data.

TD2.4 Where standard protocols are used, the tester shall confirm that it is not possible to perform downgrade attacks, and that the system is not configured to permit use of algorithms or key sizes that do not meet the requirements of strong cryptography. The tester shall perform a public vulnerability search on the protocol and version implemented and confirm that there are no exploitable vulnerabilities in the implementation.

TD2.5 For each secure channel implementation, the tester shall detail how MITM and replay attacks are prevented. At a minimum, this shall require the use of unique keys for each secure channel session.

TD2.6 Where TLS is used for the secure channel, the tester shall note:

- The version of TLS used
- Whether the TLS code is implemented in the PIN CVM application itself, or whether the platform is relied upon to establish this connection. Where the PIN CVM application implements the TLS connection itself, the tester shall note the source and version of the TLS code used.
- The cipher suites supported for use;
- What method is used to enforce only authentic connections and use of authentic certificates (e.g., "certificate pinning" or similar)

TD2.7 The tester shall confirm that no secret or sensitive data is transferred between the devices prior to the establishment of the secure channel. This must include any aspects of the provisioning process assessed under the Secure Provisioning requirement.

TR D3: Payment and Customer Data Correlation

***PIN* data must be kept isolated from other customer identifiable data, such as e-mail addresses, phone numbers, etc. Where such data is maintained for the purposes of providing customer receipts electronically, the systems that store the customer data must be logically isolated from the systems that perform payment processing.**

Guidance

Customer data must be kept separated from payment data to increase the difficulty of any attacks that collect partial payment data and attempt to correlate this against other stolen data to derive a more complete set.

Storage areas for customer data must be logically isolated from the back-end processing environments, using different network segments with access controls that prevent direct connections between the two areas.

TD3.1 The tester shall confirm and document all customer data that is processed and maintained/stored by the system.

TD3.2 The tester shall detail how this customer data is provided to the system: through the PIN CVM application, through a direct relationship between the customer and system provider, or any other means.

TD3.3 The tester shall verify that customer data entered into the PIN CVM application is cleared after being sent to the back-end processing systems and that customer data is never stored on the merchant device.

TD3.4 The tester shall confirm that cardholder data decryption must only occur in the back-end processing environment.

TR D4: Back-end Systems: Anomaly Detection

The system vendor must implement methods to allow for the detection of potentially fraudulent activity and payments and determination of common elements within such activity.

Guidance

Anomaly-detection systems must monitor multiple data points for determining patterns in potentially fraudulent activity. These may include, but may not be limited to, data such as merchant ID, SCRP ID and keysets, PIN CVM application ID and keysets, geographical location of the merchant, merchant phone number and (e-mail) addresses, merchant relationships with other merchants who may use the system, historical data about merchant operations such as chargebacks, merchant type, transaction volume and velocity, etc.

TD4.1 The tester shall detail how the anomaly-detection system is implemented and maintained by the vendor. This must include how potential fraudulent activity is escalated and what the response to such activity may be.

TD4.2 The tester shall detail what methods are used to correlate different fraudulent attempts or activities, in an attempt to isolate commonalities. Where common data points are not used (e.g., geolocation of the merchant) justification must be provided for why this does not reduce the overall security of the system. At a minimum, data points should include:

- Merchant level transactional velocities that are statistically inconsistent with historical transaction volumes associated with PIN-based transactions
- Anomalous merchant activity related to area of geographical use inconsistent with historical activity associated with PIN-based transactions
- Individual PAN usage velocities for PIN-based transactions that may be associated with probing or testing detection capabilities in an attempt to circumvent such controls
- Suspicious activity associated with multiple transactions originating from individual PANs for PIN-based transactions within time frames inconsistent with merchant geographical locations
- Signals associated with cardholder/merchant collusion associated with PIN based transactions

In all cases, it is not sufficient for the anomaly-detection mechanism to rely on attestation data from the COTS platform – it must always include analysis and consideration of merchant and transaction-based data that is separate from the technical data collected by the monitoring system.

TD4.3 The tester shall confirm that there is a minimum time for responses to potentially fraudulent activity, and that this does not exceed 48 hours.

TD4.4 The tester shall confirm that the anomaly-detection system operates “out of band” – that is, it must monitor transactions but not be an integral part of payment processing.

TR Module 5: Back-end System Security Requirements

TR E1: Security of Back-end Systems

All back-end systems that provide security services to the PIN CVM application system must be designed, implemented and maintained securely.

Guidance

Back-end systems include the payment processing environment, as well as the monitoring systems. PCI DSS compliance must include the PIN acceptance solution in the scope of assessment, including, but not necessarily limited to, all equipment, software, procedures and personnel.

The monitor environment must be assessed to the requirements contained in Appendix A: Monitoring Environment Basic Protections, if no PAN is present in the environment.

If PAN or SAD is present anywhere in the back-end monitoring environment, a full PCI DSS plus DESV Assessment is required.

TE1.1 The tester shall obtain and review the Attestation of Compliance (AoC) outlining compliance of the vendor environment to the PCI DSS requirements. This AoC must cover the scope of the back-end attestation/monitoring environment and payment processing environment (although it may not cover non-payment-related functions such as the monitoring system).

TE1.2 Where the monitoring system is included in the scope of the PCI DSS assessment, the tester shall confirm that the monitoring environment has been assessed to the additional controls outlined in Appendix A3 of PCI DSS, "Designated Entities Supplemental Validation."

TE1.3 Where areas of the vendor system are not covered by the AoC (for example, because they do not store, process or transmit account data), but are considered important for the security of the PIN acceptance system, the tester must ensure that these are covered under testing as required by Appendix A: Monitoring Environment Basic Protections.

TE1.4 The tester shall confirm that the AoC was signed by a PCI SSC-qualified QSA within the last 12 months.

TE1.5 The tester shall confirm that the back-end monitoring systems have been assessed to the requirements contained in Appendix A: Monitoring Environment Basic Protections. If the assessment has been performed by a separate entity, the tester shall have access to the report output and shall validate that it included in scope all areas of the back-end monitoring systems and that it was performed by an approved entity.

TR E2: PCI PIN Compliance

The back-end processing environment must have been assessed to be compliant to the PCI PIN Security Requirements by an approved independent auditor. Compliance must include the systems and processes used for the COTS PIN acceptance.

Guidance

The scope of any PCI PIN audit must include the PIN on COTS PIN methods and equipment. It is understood that use of these systems may result in some areas of non-compliance due to required changes to the PCI PTS SCRP and entry of PINs on non-PTS-approved systems, but such instances should be noted and may be dismissed.

It is not acceptable to use previous PCI PIN audit results that did not include the PIN on mobile systems as the sole evidence for compliance to this requirement.

TE2.1 The tester shall confirm that the back-end payment processing system has been assessed as compliant to the PCI PIN requirements by a PIN auditor approved by one of the PCI SSC brands.

TE2.2 The tester shall confirm that the scope of this audit included all relevant systems and components used for the PIN acceptance on mobile devices. Where the audit was performed prior to the use of these systems, a finding of "not compliant" must be applied to this requirement.

TE2.3 The tester shall confirm that all key injection systems used for installing cryptographic keys into the SCRP are included in the scope of the PCI PIN audit and have been confirmed as compliant to the requirements of Normative Annex B of that standard.

TR Module 6: Card Reader Security Requirements

TR F1: Secure Card Reader

The system must require the use of a PCI PTS-approved SCRP for account data entry.

Guidance

For compliance to these requirements, SCRPs used must be an approved PCI PTS device that is listed on the PCI SSC Approved Device website with an SCRP Approval Class.

TF1.1 The tester shall confirm that the only method of entry for the account data read from chip-based transactions is an approved PCI PTS device that is listed on the PCI SSC Approved Device website with an SCRP Approval Class. The tester shall list all SCRPs used for the acceptance of card data by the system and shall provide the PCI PTS approval number for each.

TF1.2 For each of the SCRP devices used by the system, the tester shall review the Security Policy document and confirm that the cryptography used by the SCRP, as assessed throughout this standard, is listed within the document. Where this cryptography is not listed in the security policy, compliance to this requirement must involve re-assessment of the changes to the SCRP by an approved PCI PTS laboratory.

TF1.3 For each of the SCRP devices used by the system, the tester shall review the Security Policy document and confirm that the SCRP was approved for use with PINs and to provide PIN translation. Where these functions are not listed in the security policy, compliance to this requirement must involve re-assessment of the changes to the SCRP by an approved PCI PTS laboratory.

Appendix A: Monitoring Environment Basic Protections

Please see the Monitoring Environment Reporting Template in the *PCI Software-based PIN Entry on COTS Program Guide.*

Appendix B: Software Tamper-responsive Attack Costing Framework[1]

There are differences between a hardware-based tamper-responsive system and a software-based tamper-responsive system that must be taken into consideration as we look at the attack-costing framework. The key differences are:

1. **Physical, Attacker Present vs. Non-Physical, Attacker Remote**

   Attacking a hardware tamper-responsive system such as a PCI PTS POI device generally requires an attacker to be physically present with the attack target. Hardware security controls are implemented to both detect and respond to a physical attack against the system. Once detected, the typical response of a hardware tamper-responsive system is to delete all sensitive assets with no means to recover.

   Attacking a software-tamper-responsive system, on the other hand, does not generally require the attacker to be physically present with the attack target. While some attack vectors would require physical access to the system, software attack vectors are usually executed remotely. For example, attacks may be executed with a piece of code or an application under a different execution or privilege context—e.g., an app with root privileges attacking the system in user mode. Hence, detection of the attack attempt may not always be straightforward and would rely on indirect pieces of data points. Typical detection strategies rely on anomaly-detection algorithms with the data points from the software application as input and monitored over time. Once an attack attempt is detected, the options available to respond are more varied when compared to a hardware tamper-responsive system.

2. **Standalone Detection vs. Distributed Detection**

   Detection mechanisms of a hardware tamper-responsive system are typically self-contained within the device, relying on a change in a physical property (e.g., temperature or voltage) to detect an attack. The available data for the attack detection engine is typically well defined. The decision-making process of the detection engine is therefore binary in its conclusion as to whether the system is under some form of attack.

   Software tamper-responsive systems are typically implemented with tamper-detection capabilities distributed between both the mobile application and the back-end monitoring system where the attack detection engine is located. The mobile application may also be used to gather local system data that is then sent to the back-end, where it is used by anomaly-detection algorithms to decide whether an actual attack has happened on the local system. The back-end monitoring system will then make a corresponding response decision. As a result, the detection engine has the ability to make decisions that are not as binary as a standalone detection engine.

3. **Individually vs. Collectively**

   Due to the nature of the attacks against hardware-tamper mechanisms, such attacks have to be conducted individually, one device at a time, by the attacker. Each device under attack is physically subjected to the same exploitation process in an attempt to compromise the hardware-based protection.

   On the other hand, exploits against software tamper-responsive systems may target the entire collection of similar systems using a malware or virus as the distribution medium. Hence, attack

[1] This Appendix assumes that the reader is familiar with the concepts captured within Appendix B: Physical Attack Potential Formula covered within the *PCI PTS POI Derived Test Requirements* document.

vectors for software tamper-responsive systems have an additional risk dimension of scalability where the full population of similar systems could be potential targets of the exploit.

The objective of this framework is to model attack vectors that deployed solutions may potentially encounter. The cost model will not include attack vectors that are not also considered by this standard – for example, direct hardware attack on the device during the exploitation stages.

Additional Considerations for Attack Cost Calculations

From the above differences, we derive the following factors that are considered when developing the attack cost framework for a software tamper-responsive system. The attack cost factors are grouped into Attacker Present, Attacker Remote and Back-end Monitoring:

Attacker Present

Attackers typically have full access to the PIN CVM application downloaded from the web store and the COTS device on which the application is running. With the device in front of him/her, the attacker can proceed to identify and exploit both the PIN CVM application and COTS device. Factors below consider the attack cost where an attack vector requires physical access to the PIN CVM application and COTS device:

1. **Access to the COTS Device**

   Attackers may not always have full access to the device. This attack cost factor considers attack vectors that require varying degree of access to the physical device under attack. Four types of access are identified.

   a) **Remote, no user interaction:** The attack is executed remotely on a target device and no user interaction is expected by the user for the attack to be successful. Example of such an attack vector is when malicious code is injected into the PIN CVM application.

   b) **Remote, user interaction required:** The attack is executed remotely on a target device, but user action is required on the target device to allow the attack to be successfully executed. Example of such an attack vector is when the user clicks on a malicious URL.

   c) **Local locked:** The attack is executed with physical access to the device but without requiring that the device is unlocked. Example is when the attack vector requires connecting the device via USB to a computer to initiate the attack without requiring the user to first unlock the device.

   d) **Local unlocked:** The attack is executed with physical access to the device but requires that the device be first unlocked by the user. Example is when the attacker executes a jailbreaking process on a user's phone that has been left unlocked.

2. **Equipment Required**

   Attackers may require the use of equipment in the execution of some attack vectors. The type of equipment required to execute an attack vector will determine the attack cost associated. The three levels are:

   a) **Standard/software only:** Where the attacker uses equipment that is easily obtainable from a consumer electronics store or executes the attack vector using software only. Examples are USB cables and software to spoof geolocation reporting by the phone.

   b) **Specialized:** Where the attacker uses specialized equipment that is not easily obtainable but has to be either custom built or modified from a piece of standard equipment to serve a specialized attack function. Examples are IMSI tracker and SIM card seizure tools.

c) **Chip level:** Equipment that directly attacks the chipsets on the device in an attempt to compromise sensitive data. Examples include equipment to initiate an electromagnetic fault injection (EMFI) attack on the chip.

3. Expertise to Execute the Attack Vector

Optimizing the attack vector to target the various combinations of system configurations would require different levels of expertise depending on both publicly available information and the attacker's technical understanding of the systems in question. Identified levels of expertise are as follows:

a) **Layman:** Persons without professional or specialized knowledge in a particular subject. They are unknowledgeable compared to experts, proficient or skilled persons with no particular expertise but who are capable of implementing simple steps to optimize an attack vector. 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 able to perform more complex optimization to attack vectors without direction. They have the ability and training to perform a specific task well.

c) **Proficient:** Persons who are highly competent and have the necessary ability, knowledge and skill to perform complex customization of attacks successfully. They are familiar with the security functionalities and behavior of the underlying systems.

d) **Experts:** Persons who 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.

4. Attack Time

The attack time factor is the amount of time that is required for the identification and exploitation of an attack vector. In calculating the attack cost for identification and exploitation, considerations for the other factors must be factored into the calculation of the time:

a) **Identification Costing**

As part of calculating the attack cost of attacking the solution, the lab should take into consideration reliance by the PIN CVM application on native COTS security features as well as any other security controls that the vendor has integrated into the solution like obfuscation, white-box crypto, etc.

At the same time, the lab should also take into consideration the four factors above (Scalability, Expertise to Execute the Attack Vector, Quality of Attestation Data and Knowledge of the Back-end Monitoring System) in the estimation of the attack time for the identification phase.

b) **Exploitation Costing**

In calculating the attack time exploitation cost, the operational quality of the back-end monitoring system must be factored into it. The lab will be required to have access to the back-end monitoring system as it assesses the exploitation attack time factor. This is required to ensure that the attack exploitation time takes into consideration how the back-end monitoring system will respond to the specific attack vector. The lab is expected to evaluate the operational quality of the back-end monitoring system (see below) at the same time it is making a determination of the attack exploitation time cost.

Attacker Remote

Attacker remote is when the attacker does not have the COTS device in front of him or her. As a result, the attacker may not have full access to the PIN CVM application downloaded from the web store and the COTS device on which the application is running. Factors below consider the attack cost where the attack is conducted by a remote attacker:

5. **Scalability Factor**

   While the time and resources required to identify and exploit a vulnerability may be the same for both hardware and software tamper-responsive systems, we have to take into consideration that a software attack on a solution running on a COTS platform may be scalable to impact a population of similar systems within a very short timeframe. The same exploit can be optimized to apply to different system configurations. Customization would then include consideration and identification of the deployment mechanism—e.g., malware, phishing, virus—used to distribute the exploit to the system population in question:

   a) **No customization required:** The attack vector can be applied to the entire population of system configurations.

   b) **Customized for each vendor:** The attack vector must be optimized based on the vendor/manufacturer of the solution.

   c) **Customized for each device model:** The attack vector must be optimized to work for each model of device.

   d) **Customized for each major OS release:** The attack vector must be optimized to work on each of the major OS versions supported by the solution.

   e) **Customized for each minor OS release:** The attack vector must be optimized to work on each of the minor OS version supported by the solution.

   f) **Customized for each instance:** The attack vector must be optimized for each instance of a system configuration.

6. **Expertise to Optimize the Attack Vector for Scalability**

   Optimizing the attack vector to target the various combinations of system configurations would require different levels of expertise depending on both publicly available information and the attacker's technical understanding of the systems in question. Identified levels of expertise are as follows:

   a) **Layman:** Persons without professional or specialized knowledge in a particular subject. They are unknowledgeable compared to experts, proficient or skilled persons with no particular expertise but who are capable of implementing simple steps to optimize an attack vector. 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 able to perform more complex optimization to attack vectors without direction. They have the ability and training to perform a specific task well.

   c) **Proficient:** Persons who are highly competent and have the necessary ability, knowledge and skill to perform complex customization of attacks successfully. They are familiar with the security functionalities and behavior of the underlying systems.

   d) **Experts:** Persons who 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.

7. Quality of Attestation Data

To bypass the monitoring system, the attacker has to suppress or simulate the attestation data sent by the PIN CVM application to the back-end monitoring system. Hence, the ability of the PIN CVM application to send the attestation data and the quality of the attestation data are important in determining the effort required to simulate this data to subvert the back-end monitoring system. The levels of the attestation data can be differentiated as:

a) **Low quality,** where the data is easily suppressed or can be easily simulated by another application with the intent of fooling the back-end monitoring system that the system has not been modified. Examples of such data known to be easily spoofed include but are not limited to geolocation data and the device's IP address.

b) **Medium quality,** where the data provides a high level of assurance that the information is authentic and has not been spoofed. An example might be the integrity information from a software-based protection mechanism, such as white-box cryptography solutions.

c) **High quality,** where the data cannot be easily spoofed or simulated. Examples are cryptographically based attestation data or attestation data that contains information obtained from the underlying hardware. Examples of such data include information from hardware-based modules like secure elements or security processors.

The quality of the attestation data does not apply to any individual datum, but to the sum of all the attestation data provided by the PIN CVM application to the back-end monitoring system by which attack detection decisions are made. It is the responsibility of the lab to determine the rating of the quality of the combined set of attestation data.

8. Knowledge of the Back-end Monitoring Systems

This refers to the information of the back-end monitoring systems. It includes information on its capabilities and behavior, possibly including the anomaly detection algorithms used to interpret the various attestation data that comes from the application. Identified levels are as follows:

a) **Public information** about the back-end monitoring system (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 to any customer.

b) **Restricted information** concerning the back-end monitoring system (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.

c) **Sensitive information** about the back-end monitoring system—for example, knowledge of internal design, which may have to be obtained by "social engineering" or exhaustive reverse engineering.

9. Attack Time

The attack time factor is the amount of time that is required for the identification and exploitation of an attack vector. In calculating the attack cost for identification and exploitation, considerations for the other factors must be factored into the calculation of the time:

a) **Identification Costing**

As part of calculating the attack cost of attacking the solution, the lab should take into consideration reliance by the PIN CVM application on native COTS security features as well as any other security controls that the vendor has integrated into the solution like obfuscation, white-box crypto, etc.

At the same time, the lab should also take into consideration the four factors above (Scalability, Expertise to Execute the Attack Vector, Quality of Attestation Data, and

Knowledge of the Back-end Monitoring System) in the estimation of the attack time for the identification phase.

b) **Exploitation Costing**

In calculating the attack time exploitation cost, the operational quality of the back-end monitoring system must be factored into the cost. The lab will be required to have access to the back-end monitoring system as it assesses the exploitation attack time factor. This is required to ensure that the attack exploitation time takes into consideration how the back-end monitoring system will respond to the specific attack vector. The lab is expected to evaluate the operational quality of the back-end monitoring system (see below) at the same time it is making a determination of the attack exploitation time cost.

Back-end Monitoring

The back-end monitoring system is a critical component of the overall software tamper-responsive system. This component is especially critical in a software-based PIN entry solution where depending on security and risk management policies, the monitoring system may immediately terminate the PIN entry capability of any COTS where there are signs that the device may be compromised.

10. Operational Quality of Back-end Monitoring System

It is important that the various rules used for anomaly detection, processes to update the monitoring systems, detection data from the PIN CVM application, etc. are constantly updated based on the latest information available. This will in a large part depend on proficient personnel trained to identify attack signatures and provide the relevant updates to the back-end monitoring systems.

The three levels are:

a) **Low operational quality,** where the rules, policies, processes and personnel involved in operating the back-end monitoring system are not able to demonstrate the proficiency required to ensure the timely identification of attack attempts.

b) **Medium operational quality,** where the lab was able to establish a level of comfort where the rules, policies, processes and personnel operating the back-end monitoring system understand their roles and will ensure that the majority of attack attempts are identified.

c) **High operational quality,** where the rules, policies, processes and personnel involved in operating the back-end monitoring system demonstrate a high level of proficiency that provide the assurances to the lab that any attack attempts will be promptly identified and that the system will be updated to respond appropriately to any future attempts.

It is expected that the lab provides an initial identification of the operational quality of the back-end monitoring system as the solution is first presented for testing. Since the solution would not have been in production, the initial identification costing of the quality would only provide a baseline cost. Subsequent periodic testing will then be required to determine the exploitation attack cost of the quality of the back-end monitoring system.

Given the nature and importance placed on back-end monitoring systems, vendor solutions that are rated Low either during the initial evaluation or during subsequent periodic testing will immediately fail the evaluation.

An Approach to Calculation

The section above 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 scalability for detection). 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.

Attack Factors		Range	Guidelines	
			Identification	Exploitation
Attacker Present	Access to the COTS Device	Remote, no user interaction	NA	0
		Remote, user interaction required	NA	1
		Local locked	NA	2
		Local unlocked	NA	3
	Equipment Required	Standard/software only	0	0
		Specialized	3	3
		Chip level	7	7
	Expertise to Execute the Attack Vector	Layman	NA	0
		Skilled	NA	1
		Proficient	NA	3
		Expert	NA	4
	Attack Time	≤ 12 hours	0	0*
		≤ 1 day	2	2*
		≤ 1 week	3	3*
		≤ 1 month	5	5*
		Beyond 1 month	8	8*
Attacker Remote	Scalability Factor	No customization required	1	NA
		Customized for each vendor	2	NA
		Customized for each device model	3	NA
		Customized for each major OS variant	5	NA
		Customized for each minor OS variant	8	NA
		Customized for each instance	13	NA
		Layman	NA	0
		Skilled	NA	1

Attack Factors		Range	Guidelines	
			Identification	Exploitation
Attacker Remote (continued)	Expertise to Optimize the Attack Vector for scalability	Proficient	NA	3
		Expert	NA	4
	Quality of Attestation Data (Bypassing automated monitoring)	Low	0	0
		Medium	5	5
		High	10	10
	Knowledge of the back-end monitoring system	Public	0	0
		Restricted	5	5
		Sensitive	10	10
	Attack Time	≤ 12 hours	0	0*
		≤ 1 day	2	2*
		≤ 1 week	3	3*
		≤ 1 month	5	5*
		Beyond 1 month	8	8*
Back-end Monitoring	Operational Quality of Back-end Monitoring System	Low	0*	Costing to be provided by a periodically recurring process. (See below)
		Medium	5*	
		High	10*	

* *Exploitation cost of attack time will be done in tandem with the calculation of the operational quality of the back-end monitoring system as the lab is expected to have access to the back-end monitoring system while attempting to exploit the system.*

Operational Quality of Back-end Monitoring Systems – Exploitation

Unlike a hardware-based responsive system that has limited opportunity to be updated in the field, a software-based tamper-responsive system with a back-end monitoring component has continued opportunity and expectation for the solution to be constantly updated and patched in response to newly discovered vulnerabilities. Hence, when evaluating the cost to exploit software-based tamper-responsive systems, it is important to also include an ongoing evaluation of the operational quality of the back-end monitoring system beyond the initial identification of its quality.

Therefore, it is expected that a periodic testing of the back-end monitoring system be conducted to ensure its operational quality is being maintained. This testing will be executed in production like the current quarterly scan requirement under PCI DSS. The objective of this ongoing assessment is to ensure operational quality for the back-end monitoring system and how it has been updated to respond to newly identified attack vectors and vulnerabilities and provide in-field update of the PIN CVM application.

Periodic In-field Test	
Vendor Expectations	**Lab Expectation**
1. Provide information on the actual adoption rate of the various supported COTS platform. 2. Provide information on how the vendor has kept up with the latest attack vector in the industry. 3. Be prepared to provide supporting information from the monitoring system of the test results.	1. Provide tester with vendor information on the various attestation data and events that would result in back-end monitoring system alerts. 2. Sign up for a merchant account using different phones and potentially from different geolocations, initiate test to attempt bypass the controls. 3. Obtain monitoring system event log information from the vendor. 4. Taking the actual merchant distribution of the various COTS platform into consideration, calculate the exploitation cost by evaluating the operational quality of the back-end monitoring system.

An approach such as this cannot consider every circumstance or factor but should give a better indication of the attack potential. Other factors, such as the reliance on unlikely chance occurrences, 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 all the factors described above.

This information is best summarized in a table containing all the items described above.

Attack Examples

Attack Example 1 – Remote Exploitation of Known, Unpatched Vulnerability	
Conditions	**Assumptions**
1. Device uses an Android application with obfuscated native library. 2. The obfuscation is sufficient to confuse IDA Pro and require manual reverse engineering but is not applied at the OS/app level for interface to the touch-event data. 3. The library is designed such that highly sensitive code is unreadable without having retrieved some information from an online host. 4. However, the touch data from PIN entry can be hooked if root access is available on the device. 5. This attack does this through exploitation of an unpatched RCE vulnerability on the device to gain execution. 6. A further privilege escalation vulnerability is then exploited to then grant root access to the installed code	1. Vulnerabilities are known to affect all versions of Android v6 and below but require user to “click to install” an application and accept elevated privileges. 2. A static keypad is used for the PIN entry interface. 3. The monitoring system is not able to consistently detect rooted COTS device.

Attack Factor		Range	Identification	Exploitation
Attacker Present	Access to the COTS Device	Remote, user interaction required	NA	1
	Equipment Required	Standard/software only	0	0
	Expertise to Execute the Attack	Skilled	NA	1
	Attack Time	NA	NA	NA
Attacker Remote	Scalability Factor	No customization required	1	NA
	Expertise to Optimize the Attack Vector for Scalability	Layman	NA	0
	Quality of Attestation Data (bypassing automated monitoring)	Low	0	0
	Knowledge of the Back-end Monitoring System	Public	0	0
	Attack Time	≤ 12 hours	NA	0*
		≤ 1 month	6	NA
Back-end Monitoring	Operational Quality of Back-end Monitoring System	Low	0*	Costing to be provided by a periodically recurring process.
	Attack Potential per Phase		7	5
	Total Attack Potential		**12**	

Attack Example 2 Remote Exploitation of Known, Unpatched Vulnerability Requiring Customization	
Conditions	**Assumptions**
Conditions are the same as Example 1 but now the attack uses a vulnerability that must be customized for each minor OS version or include additional code that brute forces the ASLR to "self-customize" for each instance.	1. Vulnerabilities are known to affect all versions of Android v6 and below but require user to "click to install" an application and accept elevated privileges. 2. A static keypad is used for the PIN entry interface. 3. The monitoring system is normally able to detect other applications running as root, but a method around this detection has been determined (and therefore a determination of "medium" has been applied to the monitoring system).

Factor		Range	Identification	Exploitation
Attacker Present	Access to the COTS Device	Remote, user interaction required	NA	1
	Equipment Required	Standard/software only	0	0
	Expertise to Execute the Attack	Skilled	NA	1
	Attack Time	NA	NA	NA
Attacker Remote	Scalability Factor	Customized for each minor OS variant	8	NA
	Expertise to Optimize the Attack Vector for Scalability	Expert	NA	4
	Quality of Attestation Data (bypassing automated monitoring)	Low	0	0
	Knowledge of the Back-end Monitoring System	Public	0	0
	Attack Time	≤ 12 hours	NA	0*
		Beyond 1 month	8	NA
Back-end Monitoring	Operational Quality of Back-end Monitoring System	Low	0*	Costing to be provided by a periodically recurring process.
		Attack Potential per Phase	16	6
		Total Attack Potential	**22**	

Attack Example 3 Create Custom OS / Root own Phone for Collecting PINs	
Conditions	**Assumptions**
1. Conditions involve a path to previous examples, but now the root control of the OS needs to be performed by installing a customized version of Android/iOS on the device, which provides the hooks to capture the PIN data. 2. This means that the attack is not scalable – it requires full physical and logical access to the attack device and must be developed and installed on that device only	1. A static keypad is used for the PIN entry interface. 2. Such an attack is only necessary when the monitoring system is of a high quality. 3. Development of the attack is assumed to require creation of an OS version that does not activate the monitor triggers, giving that a value of "high" as well. 4. Where the attack can be developed by "simply" creating any custom OS variant, the monitoring system triggers must be assigned a "low" value.

Factor		Range	Identification	Exploitation
Attacker Present	Access to the COTS Device	Local unlocked	NA	6
	Equipment Required	Standard/software only	0	0
	Expertise to Execute the Attack	Expert	NA	4
	Attack Time	≤ 12 hours	NA	0*
		Beyond 1 month	8	NA
Attacker Remote	Scalability Factor	Customized for each instance	13	NA
	Expertise to Optimize the Attack Vector for Scalability	Expert	NA	4
	Quality of Attestation Data (bypassing automated monitoring)	High	10	10
	Knowledge of the Back-end Monitoring System	Public	0	0
	Attack Time	≤ 12 hours	NA	0*
		Beyond 1 month	8	NA
Back-end Monitoring	Operational Quality of Back-end Monitoring System	High	10*	Costing to be provided by a periodically recurring process.
	Attack Potential per Phase		49	24
	Total Attack Potential		**73**	

Attack Example 4 Remote Side-channel Extraction of Keys	
Conditions	**Assumptions**
1. The COTS systems are using a secret/private cryptographic system on the COTS device where the key storage is vulnerable to some remote side-channel extraction – such as through a cache timing or speculative execution timing vulnerability. 2. The attack must be customized for each minor OS version but can be deployed through JavaScript on the COTS browser (so no user interaction is required). 3. The attack does not interact with the monitor, so monitor operational quality is assigned a "low" value. 4. Triggers must be assigned a "low" value.	None

Factor		Range	Identification	Exploitation
Attacker Present	Access to the COTS Device	Remote, no user interaction	NA	0
	Equipment Required	Standard/software only	0	0
	Expertise to Execute the Attack	Skilled	NA	1
	Attack Time	≤ 12 hours	NA	0*
		Beyond 1 month	8	NA
Attacker Remote	Scalability Factor	Customized for each instance	13	NA
	Expertise to Optimize the Attack Vector for Scalability	Expert	NA	4
	Quality of Attestation Data (bypassing automated monitoring)	Low	0	0
	Knowledge of the Back-end Monitoring System	Public	0	0
	Attack Time	≤ 12 hours	NA	0*
		Beyond 1 month	8	NA
Back-end Monitoring	Operational Quality of Back-end Monitoring System	Low	0*	Costing to be provided by a periodically recurring process.
		Attack Potential per Phase	29	5
		Total Attack Potential	**34**	

Appendix C: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms

The following are the minimum key sizes 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 SSC payment brand relevant transactions:

Algorithm	RSA	Elliptic Curve	DSA	AES
Minimum key size in number of bits:	2048	224	2048/224	128

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 key sizes by row are considered equivalent.

Algorithm	RSA	Elliptic Curve	DSA/D-H	AES
Minimum key size in number of bits:	2048	224	2048/224	–
Minimum key size in number of bits:	3072	256	3072/256	128
Minimum key size in number of bits:	7680	384	7680/384	192
Minimum key size in number of bits:	15360	512	15360/512	256

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.

For implementations using Diffie-Hellman (DH) or Elliptic Curve Diffie-Hellman (ECDH):

- **DH 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 shall generate a private key *x* and a public key *y* using the domain parameters (*p, q, g*).
- **ECDH 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 FIPS186-4). The elliptic curve specified by the domain parameters must be at least as secure as P-224. Each entity shall generate a private key *d* and a public key *Q* using the specified elliptic curve domain parameters. (See *FIPS186-4* for methods of generating *d* and *Q.*)
- Each private key shall be statistically unique, unpredictable and created using an approved RNG as described in this document.
- Entities must authenticate the DH or ECDH public keys using DSA, ECDSA, a certificate or a symmetric MAC (see *ISO 16609 – Banking – Requirements* for message authentication using symmetric techniques). One of the following should be used: MAC algorithm 1 using padding method 3, MAC algorithm 5 using padding method 4.

Where Diffie-Hellman key-agreement processes are used, the tester shall ensure that this is implemented securely against known attacks and ensure that the value of "g" is not common across implementations or instantiations of the PIN CVM application.

PIN encryption within the SCRP must adhere to ISO 9564 Part 2 approved algorithms for PIN encryption. If TDES is used, keys must have a minimum strength of 112 bits.

TLS implementations must prevent the use of cipher suites that do not enforce the use of cryptographic ciphers, hash functions and key lengths as outlined in this Appendix.

For hash algorithms used for authentication or security purposes, only the following algorithms and associated bit lengths are permitted:

Algorithm	Length
SHA2 family	>255
SHA3 family	>255

Appendix D: Security Evaluation Laboratory Requirements

This appendix focuses specifically on the equipment, skills and experience requirements necessary to undertake a security evaluation against the Software-based PIN Entry on COTS Requirements.

Equipment

The laboratory must have possession of, and/or access to, all equipment (physical equipment and software tools) necessary to execute all test activities. Laboratories are expected to have, on site, most equipment defined by PCI PTS as: "standard," "specialized" and "bespoke."

The following list illustrates the types of equipment a laboratory should have, of an adequate quality and sufficient volume, for the most commonly performed test activities.

- PCs/workstations, data storage and data backup facilities
- Virtualization environments for dynamic analysis
- Interfaces for equipment and devices under test, for example: cables, communications software, smart card reader, etc.
- Tools for code disassembly and decompilation
- Environmental chambers for variable temperatures, etc.
- Electronics testing tools, for example: variable voltage supplies, signal generators, amplifiers, digital storage oscilloscope, etc.
- Signal acquisition equipment, for example: antennae, probes, EM coils, microphones, etc.
- Signal analysis and signal processing software, capable of filtering, compressing, synchronizing or otherwise effectively operating on acquired signals
- Side-channel analysis test tools including effective UIs and configurable collection and analysis components
- Fault injection resources such as perturbation source and glitch control tools for these sources, for example: pulse generators, lasers, etc.
- NRNG analysis software
- Tools and interfaces capable of communicating with devices over various protocols to investigate logical anomalies and error-exploitation attacks such as fuzzing, for example: protocol analyzers and sniffers, configuration and analysis software, test scripts, etc.
- Use of web proxies and traffic interception tools for client server traffic analysis

The laboratory must have effective arrangements for utilizing vendor test tools when necessary, for device-specific testing.

Skills and Experience

The laboratory's personnel must have, collectively, the necessary skills and experience to execute all test activities. PCI SSC expects personnel to include individuals having expert-level expertise for subjects directly related to evaluations. This expertise must include both the knowledge needed to perform vulnerability analysis and strong practical capabilities for applying that knowledge in "hands-on" testing. Laboratory personnel must also have, overall, thorough knowledge in diverse subject areas not directly associated with evaluations.

The following list outlines the types of subject areas (capabilities, knowledge and skills) laboratory personnel should have:

- Familiarity with mobile execution environments and their security models; for example, Android and iOS
- Familiarity with software protection tools and their analysis, including obfuscation and white-box cryptography
- Familiarity with hooking techniques
- Device physical components: structure and materials, how multiple components combine to realize security objectives and how flaws can be identified and/or exploited
- Device security-critical components design and functionality, such as (but not restricted to): keypads, displays and cases
- Schematic representations of hardware and logic, for example, Gerber files and block diagrams; and the ability to detect flaws or weaknesses from these
- Device logical components: architecture functions, of how multiple components interact to realize security objectives and how flaws can be identified and/or exploited
- Formulation and analysis of all aspects of monitoring, penetration or modification attacks; for example, but not restricted to: device modification (electronic, mechanical and chemical), side-channel analysis and glitching/fault injection
- Operating evaluation tools for activities such as (but not restricted to): virtual environments, failure analysis, signal processing, vulnerability scanning, communications interfaces, fuzzing, side-channel acquisition, side-channel analysis, OS testing, source code analysis, NRNG testing, TCP/IP testing, mechanical lab equipment
- Source code review, including the detection of vulnerabilities
- Programming languages such as C, and other programming languages that may be in scope of a device evaluation
- Assembler language and security-relevant behaviors of compilers/interpreters. Security-relevant characteristics of OSs and OS resource management including: I/O, memory management, displays, prompts, keyboards and readers
- Linux-based OSs, and any other OSs that may be in scope of a device evaluation, including proprietary OSs, application separation, access permissions and application loading and deletion
- Cryptology, key management, key loading, firmware loading, PIN encryption, with regard to cryptology and the EMV application layer
- Transport layer security in devices, for example: Bluetooth, Ethernet, smart cards, USB, etc.
- Operating vendors' development test tools as part of an evaluation
- Entire device lifecycles and the relevance of fraud models and threats
- Hardware security
- Machine learning, including adversarial techniques

Documentation and Materials

Documentation and samples required for the assessment of the PIN acceptance solution should be agreed between the application vendor and the evaluation laboratory. The laboratory may need to request additional evaluation material when necessary. Examples of vendor-provided materials are as follows:

- Supporting documentation that will aid the evaluation, such as block diagrams, schematics and flowcharts
- Any necessary hardware and software accessories required to perform software-based PIN CVM application functionality
- Documentation that relates to the development process that can be audited by the laboratory. Examples of such documentation include:
  - Software quality procedures
  - Documentation and software control procedures
  - Change forms
  - Change-control logs
  - Change records
- The user guidance provided by the vendor that includes a description of system level security mechanisms that mitigate vulnerabilities that may be present in the PIN CVM application and mobile device

Laboratory Evaluation of Product and Monitoring Environment Basic Protections

The magnitude and scope of the evaluation of the PIN acceptance solution will vary depending upon the solution architecture. The evaluation includes a threat and vulnerability assessment of identified security assets. The vulnerability analysis should include currently known logical and physical attacks (threats) that are applicable to the PIN acceptance solution. The laboratory performs the required evaluation and generates an evaluation report documenting the results.

Evaluation may include physical testing of product samples, assessment of the design documentation or auditing of the vendor's development processes. See section "Documentation and Materials."

Laboratory Review of Software Protection Tools

When the PIN CVM application is protected using software protection tools, the laboratory should apply evaluation techniques that are appropriate to those tools and try to bypass them or determine their effectiveness in protecting PIN CVM application assets, including:

- De-obfuscation of protected code
- Bypassing any anti-tamper protections
- Identification of control flows and data flows
- Direct key recovery from memory dumps
- Bypass of applied "node locking" or Device binding such that the PIN CVM application cannot be executed from different platforms than intended

- Protection of cryptographic keys or cryptographic APIs (against misuse)
- Analysis of white-box cryptography-protected implementation (to obtain the WBC key)

If the tools have been provided by a third party and applied to the PIN CVM application by the application vendor, the laboratory may establish whether there is any available and applicable assurance for the tool.

Note: *The laboratory will have to be able to justify any re-use of assurance within the context of the PIN CVM application evaluation if the tool has a number of parameterization options.*

Laboratory Review of Source Code

All (source) code within scope of the evaluation must be made available to the vendor's chosen laboratory, either at the approved laboratory's premises or by secure remote access. Code review may be performed at the vendor location, but the code should still be available at the approved laboratory premises for the duration of the evaluation.

If the PIN CVM application solution delegates functionality to open source libraries, these must also form part of the code review.

If the (third-party) vendor of such tools can provide independent assurance for their integration with the PIN CVM application, it may be useful for the laboratory assessment; however, if the tool is configurable, the final assurance will be qualified by the level of configuration of the tools rather than the assurance provided by the tool itself.

Laboratory Review of Source Code Development Processes

The integrity of the PIN CVM application solution software development process contributes to the assurance provided by the final product. The laboratory will establish that the vendor has implemented a software development methodology that includes quality controls and change controls.

Server-based Security Mechanisms

The PIN CVM application relies on server-based mechanisms as part of its security support. For example, rooting detection may be implemented by a server that samples the runtime environment of the mobile device, or WB components may be implemented at the server. It is expected that the laboratory includes a full analysis of these mechanisms as they provide a major part of the system security.

Residual Risk

If at the end of the evaluation there remain any unevaluated proprietary components that contribute to the PIN CVM application's solution security functionality, then these will be listed as a Residual Risk to the system.

Deliverables from the Laboratory

1. The security evaluation report of both the PIN CVM application and the Monitoring Environment Basic Protections and its associated user guidance
2. A Residual Risk Analysis outlining any security user guidance

Evaluation results

Evaluation reports should be constructed as follows:

- A description of what was provided to the evaluation laboratory by the vendor
- A description of any existing assurance that was used to support the evaluation
- Whether a full, delta or other type of evaluation was performed
- A description of the product architecture with accompanying diagrams
- For the PIN CVM application on the COTS device
  - A vulnerability analysis of the application security functionality
  - Detail of any residual vulnerabilities
- Sufficient reporting of penetration testing to prove that the tests were completed, as appropriate, in order to reach the conclusions on the assurance level
- A description of any restrictions that were placed upon the laboratory by the vendor and prevented the evaluation from being fully white-box – for example, restricted access to source code or documentation.
- Conclusions of the evaluation should be modelled on the PCI PTS assurance rating methodology where vulnerabilities are rated in terms of their attack potential.
- The report should contain a summary identifying:
  - A list of identified highest-risk, complete, attack paths,
  - The associated attack potentials
  - An assessment of any vendor provided user guidance
  - A calculation of the overall assurance provided by the PIN CVM application

Appendix E: 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 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.

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 length	9	[4]
Overlapping 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 SP 800-22r1a 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 $[\log_2 n] - 5$. 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: https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software
  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: https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software.
  Internet Archive:
  http://web.archive.org/web/20150103230340/http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html