Payment Card Industry (PCI)
Mobile Payments on COTS

Security and Test Requirements

Version 1.1

November 2024

Document Changes

Date	Version	Description
Nov 2022	1.0	Initial public release.
Jan 2023	1.0.1	Errata release. Various typographical errors fixed. Included missing "not" into Requirement 1D-4.3 to align with intent and testing instruction. Updated Applicability Matrix (Table 2) to align with Domain 4 scoping statements including A&M Service Providers.
October 2024	1.1 (RFC Draft)	Update for stakeholder feedback for next version.
November 2024	1.1	Updated based on stakeholder feedback

Contents

Introduction

This document, the *PCI Mobile Payments on COTS* (MPoC) standard (hereinafter referred to as the "MPoC Standard" or "standard"), defines security requirements, test requirements, and guidance for entities involved in the development, deployment, and operation of merchant operated mobile payment acceptance solutions that use COTS devices.

Solutions that do not use COTS devices (as defined in this standard), use devices that are intended for integration into another product (such as a bare-board computer), are not considered by this standard. Use of MPoC Solutions in environments that are not merchant attended is considered out of scope of this standard, and any decisions in this regard are left to the compliance-accepting entity.

Use of MPoC implementations for non-payment acceptance, such as for transit validation where payment is processed by another system, is possible. MPoC implementors and deployers are encouraged to reach out to their compliance-accepting entities for any further details regarding deployment rules.

The security requirements described in this document provide a security framework to protect the confidentiality and integrity of sensitive payment information captured and processed in MPoC Solutions. The test requirements outlined in this document provide details of the testing processes performed by the evaluation laboratories as part of the validation testing of the solutions.

The MPoC standard allows for an optional modular approach to the development of a mobile payment solution. An MPoC Solution is considered a monolithic solution if the MPoC Solution does not use or integrate any other listed MPoC Products, and instead is assessed as a complete implementation. Alternatively, an MPoC Solution may be considered a composite solution if it does use and/or integrate one or more listed MPoC Products. The following bullets provided below outline the different types of MPoC Product that are supported by this standard for separate validation and listing:

- **MPoC Software:** All software that implements the base functionality required by the MPoC Solution, including the functionality for accepting account data (optionally including the cardholder PIN) on COTS devices. The MPoC Software must implement at least one form of COTS-native account data entry, either COTS-native NFC or COTS-native PIN entry. The MPoC Software scope also includes the attestation components, back-end functionality, and any APIs offered. The back-end may optionally also include the payment processing environment and functionality.
  MPoC Software may rely on hardware functions or systems, such as those provided by the COTS platform(s) or back-end HSMs. MPoC Software that is separately validated and listed on the PCI SSC website is considered an MPoC Software Product.

  An MPoC Software product may include listing of both MPoC SDKs and MPoC Applications.
- **MPoC Service:** An operational component of an MPoC Solution that implements the operational aspects of an MPoC Software implementation. An MPoC Service may reference one or more listed MPoC Software products or may itself include the aspects of an MPoC Software product (that is, be a monolithic MPoC Service), but does not meet the full requirements for an MPoC Solution. An MPoC Service may include an Attestation and Monitoring Service that provides A&M services, as well as a Payment Service that

provides payment processing and key management services. An MPoC Service may be listed with multiple services provided—so a single MPoC Service listing may provide both A&M Services and Payment Services.

An MPoC Service may include listing of both MPoC SDKs and MPoC Applications.

- **MPoC Solution:** The set of components and processes that supports mobile payment acceptance and protection of account data on a COTS device. At a minimum, the solution includes the MPoC Application, attestation system, and the back-end systems and environments that perform attestation, monitoring, and payment processing. An MPoC Solution may be a monolithic implementation, that does not reference any other listed MPoC Products, or it may reference and rely upon one or more MPoC Software and MPoC Service products.

  An MPoC Solution may not include an MPoC SDK as part of its listing, but it may include MPoC Applications that integrate an MPoC SDK from an associated MPoC Software or MPoC Service. An MPoC Solution may also list monolithic MPoC Applications developed as part of the MPoC Solution.

The requirements in this standard are organized in five Domains. The first two Domains cover the technical and development aspects of the software of the MPoC product (the MPoC Software, or equivalent functionality in as implemented in a monolithic MPoC Solution (Domain 1)), and the MPoC Application (Domain 2). The last three Domains cover the operational aspects of the MPoC Software, Attestation and Monitoring Service, and MPoC Solution.

- **Domain 1: MPoC Software Core Requirements:** The security and test requirements that apply specifically to the MPoC Software and MPoC Software lifecycle processes. The core Module in this Domain includes security requirements such as secure software lifecycle processes, integrity protection, sensitive information protection, and secure channels. This Domain includes optional Modules and Sections that are applicable only to MPoC Software supporting the relevant payment acceptance or cardholder verification methods (such as COTS-native NFC, or COTS-native PIN entry).
- **Domain 2: MPoC SDK Integration**: The security requirements and test procedures that apply to MPoC Applications, including those that integrate MPoC SDKs from previously assessed MPoC Software. The requirements in this Domain include the secure integration and usage of the MPoC Software, and the security of the complete MPoC Application.
- **Domain 3: Attestation and Monitoring**: The security requirements and test procedures that apply specifically to a service provider operating the back-end attestation and monitoring environment(s). This service provider is responsible for maintaining the COTS platform baseline, interpreting and responding to data collected from the COTS platforms, and the security of attestation and monitoring environment.
- **Domain 4: MPoC Software Management**: The security requirements for the operational management of software products and the management of cryptographic keys and encryption systems used in the MPoC Solution. This includes the security requirements for signing and distribution MPoC Software and MPoC Applications, and the key management for these software products as well as back-end systems. This Domain also includes the requirements for on-going management of the COTS baseline, as well as managing the security validation of any different entities involved in the MPoC Product.

  The MPoC Software Management Domain may apply to more than one entity within an MPoC Solution.

- **Domain 5: MPoC Solution**: The security requirements and test procedures that apply to MPoC Solution providers who are responsible for managing the interaction between all parties in an MPoC Solution, and ensuring that any associated MPoC Applications, which are not already listed as part of an MPoC Software Product, meet the requirements of this standard and are included as part of their MPoC listing.

Roles and Responsibilities

There are several stakeholders involved in maintaining and managing PCI standards. The following describes the high-level roles and responsibilities as they relate to the PCI Mobile Payments on COTS standard.

- **PCI SSC** — PCI SSC maintains various PCI standards, supporting programs, and related documentation. In relation to this standard, PCI SSC:
  - Maintains the Mobile Payments on COTS Security and Test Requirements (this document).
  - Maintains supporting documentation including reporting templates, attestation forms, technical frequently asked questions (FAQs), and guidance to assist entities implementing and assessing to the standard.
  - Reviews and monitors Evaluation Reports and related change submissions submitted to PCI SSC by PCI Recognized Laboratories for completeness and competency with baseline quality, security, and security testing standards.
  - Maintains engagement with the industry security evaluation stakeholders to ensure that the evolving threat landscape is effectively addressed during the evaluation work.
  - Maintains the list of approved PCI-recognized labs qualified to perform evaluations using this standard, on the PCI SSC website.
  - Maintains a quality assurance program for PCI-recognized labs.
- **Participating Payment Brands** — The participating payment brands develop and enforce their respective programs related to compliance with PCI standards including, but not limited to:
  - Requirements, mandates, and deadlines for compliance to PCI standards.
  - Which organizations are required to comply with PCI standards.
- **PCI-Recognized Laboratories** — PCI-recognized laboratories are responsible for maintaining the required knowledge, expertise, and equipment necessary to execute all test activities, and to perform the required evaluation and generate an evaluation report documenting the results. Not all PCI-recognized laboratories are qualified to perform evaluations using this standard. For further information, please consult the *Mobile Payment on COTS Program Guide*.
- **EMVCo** — EMVCo is the global technical body owned by American Express, Discover, JCB, MasterCard, UnionPay, and Visa that facilitates the worldwide interoperability and acceptance of secure payment transactions by managing and evolving the EMV Specifications and related testing processes. Adoption of EMV Specifications and associated approval and certification processes promotes a unified international payments framework, which supports an advancing range of payment methods, technologies, and acceptance environments.

Additionally, the following entities are identified for the purpose of this standard:

- **MPoC Software vendor** — The MPoC Software vendor develops, supports, and distributes an MPoC Software product. An MPoC Software vendor is responsible for ensuring that the MPoC Software they develop meets the requirements outlined in *Domain 1: MPoC Software Core Requirements*.
- **MPoC Service provider** — An MPoC Service provider is involved in the deployment and operation of services which include the operation of the MPoC Software, or support the operation of an MPoC Solution. For example, an entity operating the attestation and monitoring (A&M) service is responsible for ensuring that all requirements outlined in *Domain 3: Attestation and Monitoring* are met, and other aspects of the MPoC requirements may apply to each MPoC Service depending on the details and scope of that service.
- **MPoC Solution provider** — An MPoC Solution provider is the entity that has overall responsibility for the implementation and management of an MPoC Solution. The MPoC Solution provider is responsible for ensuring that all requirements are met, including any requirements fulfilled by other organizations on behalf of the MPoC Solution provider (e.g., MPoC Services providing attestation and monitoring). This may involve reliance on separately validated MPoC Products, such as MPoC Software or MPoC Services. In addition, MPoC Solution providers are responsible for ensuring there is a communication path to merchants using the MPoC Solution.

Glossary/Terminology

In addition to terms defined in the *PCI DSS Glossary, Abbreviations and Acronyms*[1], the terms/acronyms listed in Table 1 are used throughout this document.

Table 1: Glossary of Terms

Term	Definition
Account data	Account data consists of cardholder data and/or sensitive authentication data. See *Cardholder data* and *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).
Assets	Assets are elements of the MPoC Solution that are security-sensitive or are used to provide security to other security-sensitive elements. Examples of assets include sensitive assets such as account data, cardholder PINs, and cryptographic keys. Software may also be considered an asset if the correct operation of that software is required to provide security protection to other assets. See also *Sensitive assets*.
Asymmetric encryption	Also known as public key cryptography, asymmetric cryptosystems are based on the intractability of certain mathematical problems. 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.
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 measurements requests and thresholds implemented by the prover.
Attestation component	An element of the solution that performs attestation processing.
Attestation system	The set of components that perform attestation processing for the MPoC Solution. The implementation may be shared across different execution environments, which provides a level of validation and assurance of the execution environment in which the MPoC Application executes, providing a level of software-based tamper detection and response. Its components include the MPoC Application attestation component and the back-end attestation component. The latter works in close association with the back-end monitoring system.

[1] *https://www.pcisecuritystandards.org/pci_security/glossary*

Term	Definition
Attestation and monitoring system (also A&M system)	The combination of the Attestation System and the Monitoring System.
Attestation and monitoring software (also A&M software)	The software used to implement the attestation and monitoring functions. Attestation and Monitoring software is a required part of any MPoC Software Product.
Back-end systems	The set of systems providing the server-side functionality of the MPoC Solution. This includes monitoring, attestation and (optional) payment and PIN processing functions.
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 might be transmitted or processed (but not stored) as part of a payment transaction.
Cardholder data environment (CDE)	The people, processes and technology that store, process, or transmit cardholder data or sensitive authentication data.
Chip-based	For the purposes of this standard, chip-based payment methods include any payment method that is based on an EMV specified protocol (contact or contactless) or originates from an ISO7816-based source.
Cleartext	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 (also COTS device)	A general-purpose computing device that is not designed solely for the purposes of payment acceptance. For the purposes of this standard this includes a mobile device such as a smartphone, tablet, or a POI device, and includes enterprise devices which are not intended for purchase or use by the public.
Contactless kernel	Software that processes contactless transactions. The kernel is selected by the MPoC Application based on the characteristics of the transaction and the payment instrument—e.g., credit card—supporting the contactless transactions. The kernel contains interface routines, security and control functions, and logic to manage a set of commands and responses to retrieve the necessary data from the payment instrument to complete a transaction.[2]
COTS-based MPoC Software	The aspect of the MPoC Software that executes on the COTS device, provides local connections to external readers (PCI PTS POI or Non-PTS Approved MSRs), and any remote execution environments relied upon for COTS-based execution (such as remote kernels). Includes the MPoC SDK and MPoC Application.

[2] EMVCo (*https://www.emvco.com/*)

Term	Definition
COTS-native NFC	The subsystem in the COTS device that is part of the COTS platform used to access data, including account data, read from contactless cards or other payment instruments. The main physical components are the NFC antenna and the NFC controller.
COTS platform	The hardware and operating systems (OS) of the COTS device.
COTS platform baseline	A measurable configuration reference point of the COTS device attributes and COTS operating systems (OS) on which the MPoC Application may be executed. The COTS platform baseline is used for periodic comparative analysis by the back-end attestation system to determine changes that would impact the overall security of the COTS device to continue to process transactions.
Cryptographic material	All materials involved in the implementation of a cryptographic algorithm or process including keys, entropy seeds, nonces, and lookup tables involved in the execution of the algorithm, etc.
Data level	Used in this standard to identify when a security control needs to be applied directly and specifically to a data type, such as PANs or PINs. Requirements for encryption at the data level cannot be met through use of Secure Channels alone.
Deterministic Random Number Generator (DRNG)	A deterministic algorithm to generate a sequence of numbers with little or no discernible pattern in the numbers, except for broad statistical properties. Also referred to as Pseudo Random Number Generator (PRNG). See *Random Number Generator (RNG)*.
Dual control	Process of using two or more separate entities (usually persons) operating in concert to protect sensitive functions or information. Both entities are equally responsible for the physical protection of materials involved in vulnerable transactions. No single person is permitted to access or use the materials (e.g., the cryptographic key). For manual key generation, conveyance, loading, storage, and retrieval, dual control requires dividing knowledge of the key among the entities. See *Split knowledge*.
Elliptic curve cryptography (ECC)	An 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.
Entity	The term entity is used to represent MPoC Software vendors, service providers, and MPoC Solution providers, as applicable, that are undergoing the review.

Term	Definition
Environment	The systems and processes supporting one or more functionalities of the 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. See *Rich execution environment (REE)*, *Trusted execution environment (TEE)*Trusted execution environment (TEE), and *Secure element (SE)*.
Hash	A method to protect data that converts data into a fixed-length message digest. A hash is a one-way (mathematical) function in which a non-secret algorithm takes any arbitrary length message as input and produces a fixed length output (usually called a "hash code" or "message digest"). Hash functions are required to have the following properties: • It is computationally infeasible to determine the original input given only the hash code, • It is computationally infeasible to find two inputs that give the same hash code.
Hash-based message authentication code (HMAC)	A code that is produced using hash algorithms rather than a symmetric cryptographic algorithm. Defined in FIPS 198-1. See *message authentication code (MAC)*.
Integrity	Ensuring consistency of data; in particular, preventing unauthorized and undetected creation, alteration, or destruction of data.
Key block	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.
Key check value (KCV)	A value used to identify a key without directly revealing any bits of the actual key itself. Also known as key verification check.
Key generation	Creation of a cryptographic key either from a Random Number Generator (RNG) or through a one-way process utilizing another cryptographic key.
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 verification checks (KVC)	See *Key check value (KCV)*.
Key wrapping	A process by which a cryptographic key is protected in integrity, confidentiality, or both by the generation of a key block to encapsulate (encrypt) the cryptographic key material for transport or storage.

Term	Definition
Magnetic-stripe reader (MSR)	A device that is used to read magnetic-stripe cards. See also *Non-PTS approved MSR*.
Mandatory access control	Access control by which the operating systems (OS) 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., through an authorization rule enforced by the operating systems (OS) kernel.
Man-in-the-middle (MITM attack) attack	An attack method where a malicious third party interposes between two other communicating parties and modifies the data sent between them.
Merchant-attended	A deployment of a payment acceptance device or solution is merchant-attended where there is a merchant or merchant agent who is able to assist with, or provide oversight of, the payment process.
Message authentication code (MAC)	In cryptography, a small piece of information used to authenticate a message. See *Hash-based message authentication code (HMAC)*.
Mobile device	In the context of this Standard, see *Commercial off-the-shelf (COTS) device*.
Mobile Device Management (MDM)	A system for the administration of mobile devices such as smartphones, tablets, computers, laptops, and desktop computers. MDM solutions may be implemented in an MPoC Solution for the purposes of the secure distribution of MPoC Applications.
M-of-N	An M-of-N scheme is a key component or key 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 should be sufficient to ensure that no subgroup of less than m persons can gain access to enough of the components to form the key.
Monolithic Solution	An MPoC Solution that is implemented without use of any other listed MPoC Products (such as an MPoC Software Product or Attestation and Monitoring Service provider).
Mobile Payments on COTS (MPoC) Software (also MPoC Software)	All software that implements the base functionality required by the Mobile Payments on COTS solution, including the MPoC SDK functionality for accepting account data and PIN entry on COTS devices (where PIN entry is supported), API, attestation components, and back-end functionality. The MPoC Software may also include the back-end payment processing environment or functions.
MPoC Software vendor	The Entity that is responsible for the development and maintenance of the MPoC Software.
Mobile Payments on COTS (MPoC) Application (also MPoC Application)	The set of all COTS-based software deployed as part of the MPoC Solution that supports mobile payment acceptance, A&M functionality, and protection of account data on a COTS device. An MPoC Application may optionally integrate the MPoC SDK portion of a listed MPoC Software product, and/or be part of an MPoC Software Product.

Term	Definition
Mobile Payments on COTS (MPoC) Solution (also MPoC Solution)	The set of components and processes that supports mobile payment acceptance and protection of account data on a COTS device. At a minimum, the solution includes the MPoC Application, attestation system, and the back-end systems and environments that perform attestation and monitoring. An MPoC Solution must implement the acceptance of chip-based payment cards (contact and/or contactless), in addition to any other payment acceptance types that it may support.
Mobile Payments on COTS (MPoC) SDK (also MPoC SDK)	The subset of MPoC Software that implements required functionality for the payment acceptance and attestation on the COTS device and secure communication with the back-end systems.
MPoC Service	An MPoC Product that provides the operational aspects of an MPoC Software product but does not meet the requirements of a full MPoC Solution. An MPoC Service is validated to all relevant MPoC requirements, based on the functionality provided.
MPoC Service (A&M)	An MPoC Service product that provides the operational A&M aspects of an MPoC Software product.
MPoC Service (A&M and Payment Processing)	An MPoC Service product that provides both the operation A&M aspects of an MPoC Software product, as well as operational aspects required for payment processing.
MPoC Service Provider	An entity that develops, manages, and/or deploys an MPoC Service.
MPoC Software vendor	An entity developing, distributing, and supporting the MPoC Software.
MPoC Solution provider	An entity that develops, manages, and/or deploys MPoC Solutions.
MPoC Software Product	The MPoC Software as validated and listed on the PCI SSC website.
Monitoring system	Monitors and provisions security controls to detect, alert, and mitigate suspected or actual threats and attacks against the MPoC Solution.
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. Contrast with a Deterministic Random Number Generator (DRNG).
Non-PTS approved MSR	An MSR that has been validated against the requirements in Appendix F of this standard.
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. Determination of an acceptable level for obfuscation may involve assessment against the attack costing methodology of this standard.

Term	Definition
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.
Operating system (OS)	System software that manages the underlying hardware and software resources and provides common services for programs. Common OSs used in a COTS environment include, but are not limited to, Android and iOS implementations.
OS store	A digital distribution service operated by the COTS OS vendor or by the COTS device manufacturer used to distribute the MPoC Application.
Out-of-band	Communication using a means independent of the primary communications means.
Payment-processing environment	Back-end environment in which account data transferred from the MPoC Software is decrypted and/or processed.
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.
Perfect forward secrecy	Also known as “Forward Secrecy.” A protocol has Perfect Forward Secrecy if a compromise of long-term keys does not also compromise past session keys.
Protection types	The specific form of protection required for an asset, including one or more of confidentiality, integrity, and authentication.
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.
Provisioning / personalization	The process of ensuring that a specific instance of a system has the correct settings and unique data to enable its operation.
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*.

Term	Definition
PIN-processing environment	Back-end environment in which cardholder PINs are processed or translated for further processing.
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.
Rich execution environment (REE)	Refers to an execution environment where COTS device resources are shared by OS applications.
RSA	Algorithm for public-key encryption described in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at Massachusetts Institute of Technology (MIT); letters RSA are the initials of their surnames.
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 listing website.
Secure channel	A physically or logically protected connection between two points.
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 and ISO 13491.
Secure element (SE)	A tamper-resistant platform (typically a one-chip secure microcontroller) capable of hosting applications and their confidential and cryptographic data (e.g., key management) securely.
Secure reading and exchange of data (SRED)	Requirements within the PCI PTS POI Standard, detailing testing for devices that protect account data.
Security processor	Within a COTS device, a security processor is a separate processor or co-processor with its own dedicated memory running separate OS, applications and data on these processors are not accessible by the COTS device's main OS.
Sensitive authentication data	Security-related information used to authenticate cardholders and/or authorize payment card transactions. This information includes, but is not limited to, card validation verification codes/values, full track data (from magnetic-stripe or equivalent on a chip), PINs, and PIN blocks.

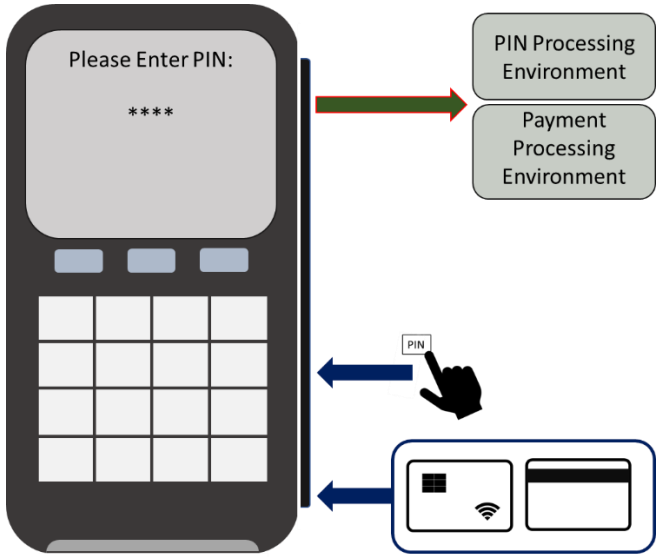
Term	Definition
Sensitive assets	For the purposes of this standard, security assets include assets that require protection by, or be used to provide protection to, the MPoC Solution (e.g., cryptographic keys, or account data).
Sensitive services	A sensitive service is any service that may affect the security of the overall system, as well as those functions that affect underlying processes that support the protection of assets—e.g., cryptographic keys and account data. Common examples are key management, modification, or update of attestation services, or remote component of contactless kernels (or other remote processing components) and cryptographic signing of assets to allow their authenticity to be verified.
Service provider	An entity responsible for deployment, operation, and management of the back-end monitoring; attestation; PIN processing; and account data processing environments.
Software-protection mechanisms	Methods and implementations used to protect against the reverse-engineering and modification of software, including, but not limited to, hooking, rooting, emulation or debugging detection, verification, and validation of software.
Software-protected cryptography	A method used to obfuscate the execution of a cryptographic algorithm in software, including the protection of the cryptographic key, with the goal of making determination of the key value computationally complex.
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.
Symmetric key	Same symmetric key that is used for encryption is also used for decryption. Also known as “secret key.”
Tamper detection	A characteristic that provides evidence that an attack has been attempted.
Tamper resistant	A characteristic that provides passive protection against an attack.
Tamper responsive	A characteristic that provides an active response to the detection of an attack, thereby preventing a successful attack.
Trusted boot	Also known as Verified Boot and Secure Boot. A cryptographic process where the bootloader verifies the integrity of all components (e.g., kernel objects) loaded during the OS startup process, but prior to loading.
Trusted execution environment (TEE)	A trusted execution environment provides hardware-based security features such as isolated execution environment for Trusted Applications. It isolates sensitive assets and code from the Rich Execution Environment (REE).
Unattended (also not merchant-attended)	A deployment of a payment acceptance device or solution is unattended where there is no merchant or merchant agent who is able to assist with, or provide oversight of, the payment process.
User interface (UI)	The set of the human-machine interfaces that allows for interaction between a person and a computerized system.

Term	Definition
vTEE	A virtualized trusted execution environment (TEE) that provides logical security features to isolate the execution environment for Trusted Applications. A type of software-protection mechanism.
White-box cryptography	A type of software-protected cryptography.

Scope of Mobile Payments on COTS Standard

In traditional merchant-facing card-based payment scenarios, account data (e.g., PAN, PIN, etc.) is entered into a device specifically designed for protecting this data, such as a PCI PTS POI approved 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.

Figure 1: Traditional PIN Acceptance Solution Overview



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). These solutions rely on a combination of mechanisms and security controls including, but not limited to, COTS device hardware, application software, and independent management and oversight of the entire process to ensure the security of the transaction and the cardholder verification data.

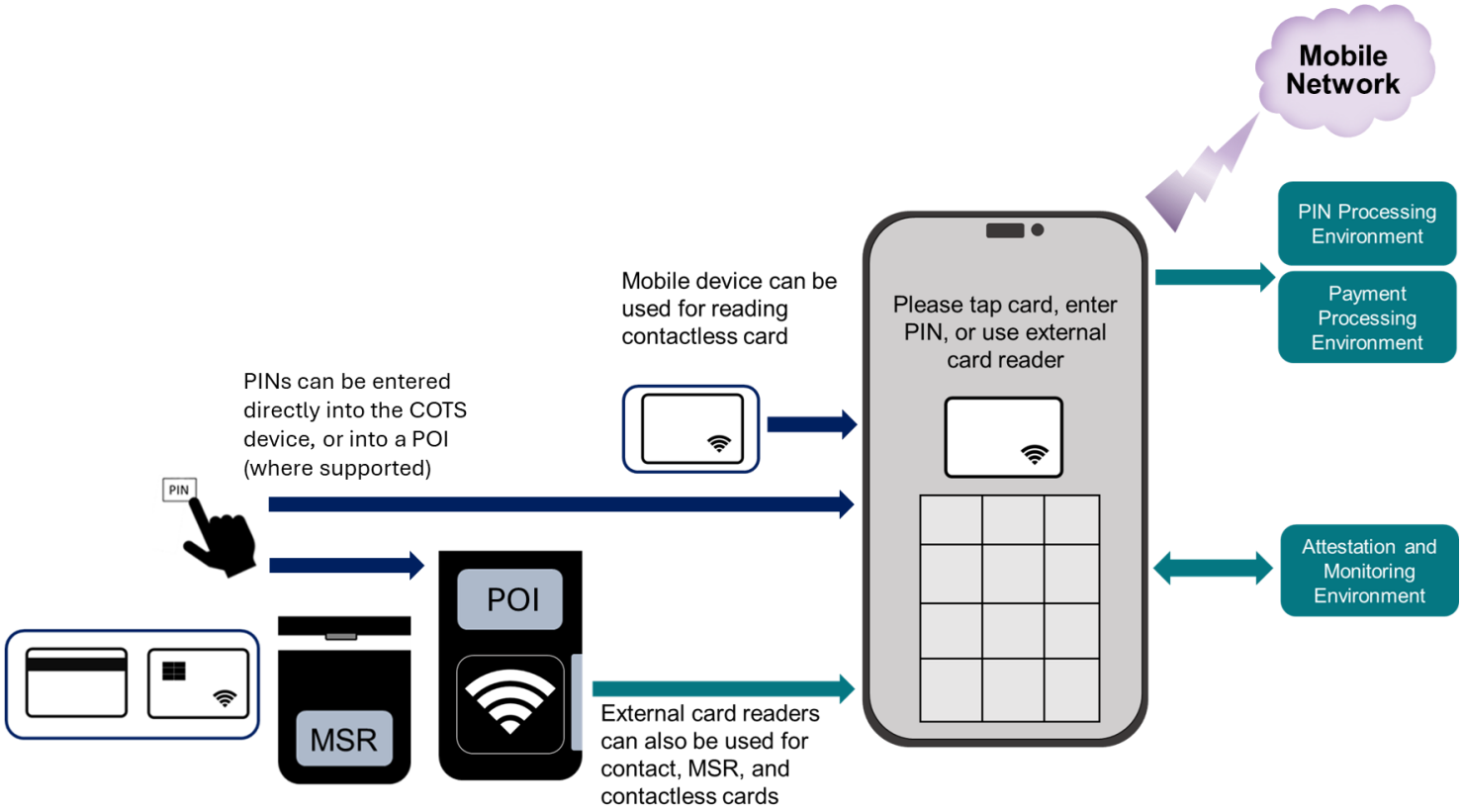
This standard is designed to allow mobile payment solutions to support multiple payment-acceptance channels and cardholder verification methods. For example, one solution could support a COTS-native NFC interface without PIN entry, while another solution could be designed to support COTS-native-NFC interfaces for contactless card entry with PIN, as well as also supporting external PCI PTS POI devices.

Mobile Payments on COTS—Payment Acceptance Channels and Cardholder Verification Methods

Figure 2 below shows the functional model of the solution that uses a software MPoC Application, with the option of additional hardware in the form of external PCI PTS POI or magnetic-stripe reader (non-PTS approved MSR), for protecting account data. Where an external PCI PTS POI is used, this can provide for reading of payment cards (both magnetic-stripe and chip-based), as well as the entry of cardholder PINs. Account data-entry methods provided for use by the POI are dependent on the PCI PTS POI approval class and functions supported.

This standard defines the requirements for assessment to validate candidate MPoC Products prior to it being listed on the PCI SSC website. For details on compliance programs outlining the use of listed MPoC Products, please refer to the relevant payment card brands.

Figure 2: Mobile Payments on COTS Solution Showing Various CVM and Account Data Acceptance Options



The standard currently supports the following payment acceptance channels:

- COTS-native NFC interface
- PCI PTS POI devices for contact, contactless, and MSR-based transactions
- Non-PTS approved MSR devices validated through the MSR appendix of this standard
- Manually entered account data and cardholder verification methods
- COTS-native PIN entry
- No CVM
- CDCVM

Support for any of the payment acceptance channels and/or CVMs described above is optional—the only requirement is that an MPoC Solution, or MPoC Software Product, must support chip-based payments and at least one COTS-native method of entry. As examples, an MPoC Solution may support COTS-native NFC and COTS-native PIN entry on the same device, it may support COTS-native NFC without PIN, or it may support account data entry through external readers only with COTS-native PIN.

However, an MPoC Solution (or MPoC Software Product) may not support only MSR or manual entry-based payments, or methods of account data entry performed exclusively through a PCI PTS POI, without any COTS-native account data-entry methods. Individual transactions are exempt from this requirement; any individual transaction may be MSR based, or exclusively use a PCI PTS POI—as long as the overall MPoC Solution (or MPoC Software Product if that is the MPoC Product under assessment) supports at least one COTS-native account data-entry method.

Where the MPoC Software supports any of the above methods it must meet and be validated to the applicable security requirements described in the optional Modules in *Domain 1: MPoC Software Core Requirements*.

Mobile Payments on COTS—Security Model

The security model of an MPoC Solution relies in large part (but not entirely) on mechanisms that support attestation and monitoring (to ensure the security mechanisms are intact and operational), detection (to notify when anomalies are present), and response (controls to alert and take action). The online nature of COTS devices provides opportunities to extend these capabilities to back-end monitoring systems.

In addition, COTS devices that have specialized hardware-based security mechanisms, such as secure element (SE) or trusted execution environment, may use these mechanisms for secure storage or processing of cryptographic material or account data, or for functions supporting the attestation of the COTS device.

There are, however, individual components of a software solution where there is limited control (e.g., the underlying COTS platform). Because 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 important for the MPoC Software to provide inherent protections that complicate reverse engineering and tampering of the code execution flow on the COTS platforms to which it is deployed. This may include, but is not limited to, protections using obfuscation of the code, internal integrity checks for code and processing flows, encryption of code segments, etc.

The architecture of an MPoC Solution relies on the following components that combine to provide for protection, attestation, detection, and response controls:

- A COTS device that is operated by the merchant to run the MPoC Application. The COTS device may have a TEE or secure element built-in, but this is not a requirement.
- MPoC Application that resides on the COTS device and:
  - Collects and passes attestation data about any attached hardware systems (such as a PCI PTS POI or non-PTS approved MSR), COTS platform, and MPoC Application to the attestation and monitoring back-end systems.
  - Contains software protection mechanisms to maintain its own integrity against attack.
  - Securely communicates account data (and other sensitive assets such as cardholder PINs, where these are supported and used) to back-end systems.
  - (Optional) a secure UI for entry of account data-entry methods (such as PIN, PAN, or security codes), and encryption of this data.
  - (Optional) a secure method for the entry and encryption of the account data acquired through COTS-native NFC read.
- Set of back-end systems that perform functions for the MPoC Solution such as:
  - Attestation and monitoring systems process attestation data from the MPoC Application and enforce pre-established security policies as well as provision security controls to detect, alert, and mitigate suspected or actual threats and attacks against the PCI PTS POI, MPoC Application, and the COTS device.
  - Processing environment that receives encrypted account data, cardholder data and, if applicable, cardholder verification method (e.g., PIN data from the PCI PTS POI).
- (Optional) PCI PTS POI or non-PTS approved MSR device that supports the solution. The PCI PTS POI is SRED-enabled, ensured to always encrypt account data output, and connected to the COTS device. The PCI PTS POI optionally provides:
  - Protection of account data
  - Translation of PIN data for forward processing
  - Secure random seeding and signing

MPoC SDKs and MPoC Applications

All software within an MPoC Software product, or software within a monolithic MPoC Solution that provides for the security of MPoC assets such as account data or cryptographic keys, must be assessed to *Domain 1: MPoC Software Core Requirements.* This includes any applicable optional Modules corresponding with the supported payment acceptance channels and cardholder verification methods (such as COTS-native PIN entry, or COTS-native NFC).

The MPoC Software is considered as two separate parts—the MPoC SDK (intended for integration into an MPoC Application) or complete MPoC Application itself, and the back-end software including the MPoC Attestation and Monitoring software. The MPoC SDK or MPoC Application execute on the COTS platform, although they may include remote execution components (such as a remote kernel).

Any MPoC Application may optionally support APIs that accept non-sensitive payment initiation data, such as an amount, from another "calling" application or function. This allows for an MPoC Solution to provide for payment initiation from applications outside of the scope of MPoC validation and listing—by providing an API through which an MPoC Application may receive non-sensitive payment initiation and response data (such as amount).

Therefore, there are three types of COTS software considered in an MPoC Solution:

1. The MPoC SDK, which is designed to be integrated into an MPoC Application.
2. The MPoC Application itself, which may or may not integrate an MPoC SDK.
3. A "calling application," which can interface to APIs exposed by the MPoC Application to initiate payment acceptance (not in scope of MPoC assessment).

For any MPoC SDK, the following methods of integrating a set of pre-developed software as a component of an MPoC Solution have been considered in the creation of this standard:

- Software that is designed and implemented as an entirely separate MPoC Application from a "calling application" executed in a completely separated execution environment (e.g., SE/TEE).
- Software that is an entirely separate MPoC Application executed in the same runtime environment as a calling application, invoked through OS APIs (e.g., Intent on Android) by that calling application.
- Software that is integrated into the MPoC Application as a pre-compiled code (e.g., library).
- Software that is integrated into the MPoC Application as source code, which is compiled with the MPoC Application (this is not a permitted implementation for an MPoC Software product).

Note: *An MPoC Application may integrate up to two MPoC SDKs.*

Due to the risk of modification of the MPoC Software, this standard does not allow for an MPoC SDK that is distributed and integrated as source code. This applies only to COTS-based MPoC software, MPoC Software implemented in back-end environments may be distributed as source code.

MPoC SDK Types and MPoC Application Implementations

An MPoC SDK, where one exists, may be implemented as one of two types:

- **An Isolated MPoC SDK**
  An Isolated MPoC SDK must provide for sufficient isolation of its memory space and be validated during the laboratory assessment that cleartext sensitive assets, such as account data or cryptographic keys, are not accessible by an MPoC Application that integrates that SDK.
- **A non-Isolated MPoC SDK**
  A non-Isolated MPoC SDK is an MPoC SDK that is unable to be validated to provide sufficient isolation to the cleartext sensitive assets from the MPoC Application.

In each case, the validation of "sufficient isolation" is based on the laboratory assessment and costing framework outlined in the MPoC standard. Assets that are already sufficiently protected, e.g., through the use of cryptography or truncation (for PANs), are not considered in the determination of an Isolated or non-Isolated SDK.

In all cases, an Isolated MPoC SDK cannot allow for sensitive cleartext data to be exposed to the MPoC Application, even if that functionality is optional or provided by some back-end function related to the MPoC Software. This is because the goal of an Isolated MPoC SDK is to prevent access to the sensitive assets—regardless of whether or not this access is intended—so that even if the MPoC Application is compromised, the sensitive assets remain secure. There can be no APIs or methods exposed that would allow for that compromised MPoC Application to gain access to those sensitive assets in cleartext.

Note: *It is not sufficient for an Isolated SDK to protect against reverse engineering only, e.g., through the use of white-box cryptography and code obfuscation alone. An Isolated SDK must prevent an MPoC Application from accessing cleartext sensitive assets, including during input/output through the physical interfaces of the COTS device (e.g., touch screen or COTS-native NFC).*

Based on these two types of MPoC SDK, the MPoC standard additionally supports the implementation of MPoC Applications in three different ways:

1. Monolithic MPoC Applications that do not integrate an MPoC SDK from a listed MPoC Software Product.

   These types of MPoC Applications must be assessed to all relevant Domain 1 requirements (only requirements covering account data-entry methods supported by the MPoC Application are assessed). Monolithic MPoC Applications are assessed to Section 2B of the MPoC standard.
2. MPoC Applications that integrate a non-Isolating MPoC SDK.

   These types of MPoC Applications integrate an MPoC SDK, from a listed MPoC Software Product, that has not been validated to provide memory and cleartext asset isolation. MPoC Applications that integrate a non-Isolating MPoC SDK must be assessed to all requirements of Domain 2 of the MPoC standard.

3. MPoC Applications that correctly integrate Isolating MPoC SDKs only.

   These types of MPoC Applications integrate an MPoC SDK, from a listed MPoC Software Product, that has been validated to provide memory and cleartext asset isolation. MPoC Applications that correctly integrate only Isolating MPoC SDKs may be assessed to the Section 2A requirements of Domain 2 of the MPoC standard only.

   If the MPoC Application does not correctly integrate an Isolating MPoC SDK, or the MPoC Application bypasses, modifies, or supplements the payment and/or security features of the Isolating MPoC SDK, then the MPoC Application must also be validated against the requirements of Section 2B of Domain 2.

An MPoC Application is permitted to integrate up to two MPoC SDKs (although this is a not a requirement, and integration of one or no MPoC SDKs is also permitted). An MPoC Application that integrates a non-Isolating MPoC SDK is always assessed against all requirements of Domain 2. Therefore, an MPoC Application that integrates two MPoC SDKs, where one is an Isolating SDK and one is a non-Isolating SDK, is required to be assessed against all requirements of Domain 2 even though one of the MPoC SDKs it integrates is an Isolating SDK.

A monolithic MPoC Application, or MPoC SDK of either type, may be implemented in part or whole outside the REE of the COTS device, e.g., as a “trustlet” executed within a TEE, which provides for the GUI and other interfaces necessary to provide the required account data inputs.

An MPoC Application may be included as part of an MPoC Software Product as well as part of an MPoC Solution. For example, an MPoC Software product may be listed as providing an Isolating MPoC SDK, a non-Isolating MPoC SDK, and/or a complete MPoC Application (which may itself integrate an MPoC SDK from the same MPoC Software Product).

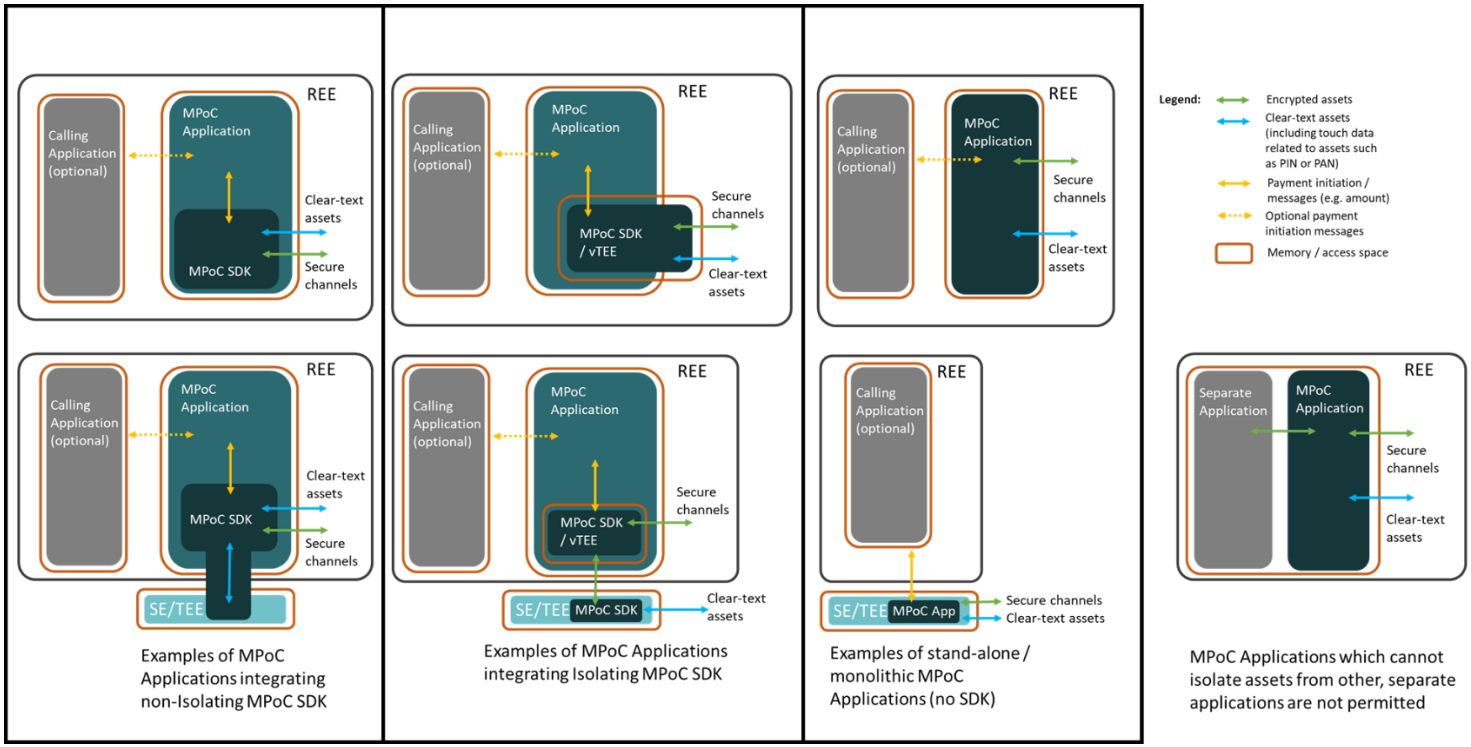
Note: *An MPoC Application listed as part of an MPoC Software Product must be developed by the MPoC Software vendor and must not integrate the SDK of any other MPoC Software Product. MPoC Applications listed as part of an MPoC Service may be developed by other entities, integrating the MPoC SDKs supported by that MPoC Service.*

MPoC Applications that access cleartext account data are to be assessed against the requirements of Domain 1, regardless of whether or not they integrate an MPoC SDK (or either type). If the MPoC Application does not access cleartext account data, and it is integrating an Isolating SDK, it may be assessed only to the requirements of Domain 2A.

An Isolated SDK may share some assets or configurations with the integrating MPoC Application, such as permissions with determine access to underlying COTS Platform systems, as long as those shared assets/configurations do not expose cleartext sensitive assets. For example, permission to access a hardware-backed keystore that contains encryption keys used to protect the sensitive assets could be shared between an Isolating MPoC SDK and the integrating MPoC Application, as long as those keys could not be used for decryption (as this would allow a compromised MPoC Application to remove the protection provided by the encryption processes).

Figure 3 illustrates the different types of MPoC Application and MPoC SDK.

Figure 3: Examples of MPoC Application Implementations

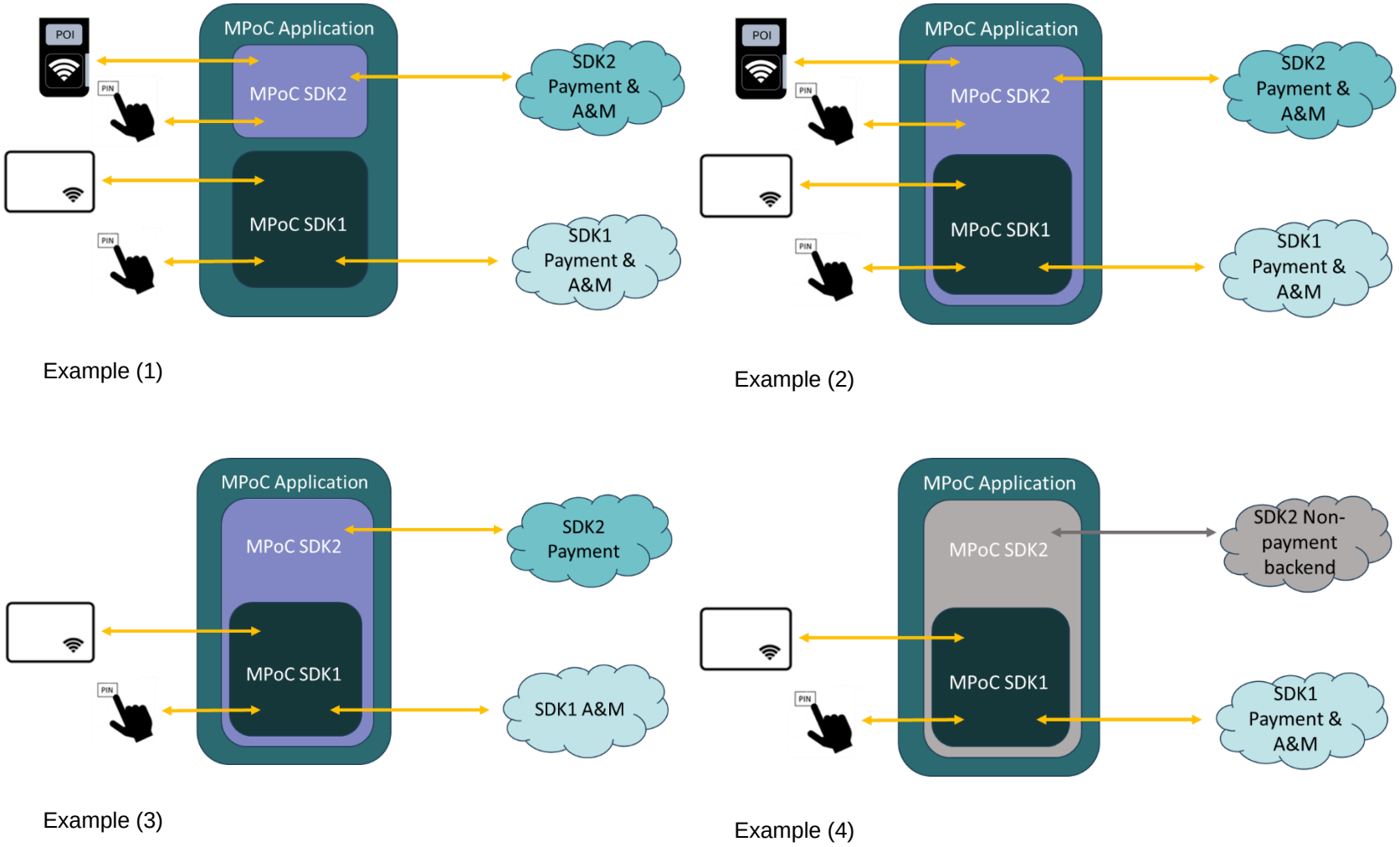


Additionally, an MPoC Application may integrate up to two MPoC SDKs, and an MPoC SDK may integrate another (single) MPoC SDK. This may be done for several potential reasons, such as:

- Adding additional MPoC-relevant payment features, such as adding support for an additional external reader to an MPoC SDK that supports only COTS-native NFC payments, which access cleartext sensitive assets.
- Adding additional payment message formatting or processing features, which may not access cleartext sensitive assets.
- Adding additional non-payment features, such as reading of transport cards through the COTS-native NFC.

Example implementations and the potential scope of assessment for these are provided below.

Figure 4: Example Implementations using Multiple MPoC SDKs



In Example (1) in Figure 4 above, an MPoC Application integrates two separate MPoC SDKs. The MPoC SDKs may be from two separate MPoC Products or may be two MPoC SDKs that are listed within the same MPoC Product. If both of the MPoC SDKs are Isolating SDKs, then the MPoC Application can be validated only through Domain 2A. If either MPoC SDK is a non-Isolating SDK, then the MPoC Application will need to be additionally validated through Domain 2B.

Example (2) is similar to the first example, where two MPoC SDKs are integrated separately, but here the first MPoC SDK (MPoC SDK1) is integrated into the second MPoC SDK (MPoC SDK2). From the point of view of the MPoC Application, it is integrating only a single MPoC SDK (MPoC SDK2). Validation requirements are the same as those in the first example.

In Example (3), MPoC SDK1 handles all account data entry and encryption, as well as management of the Attestation and Monitoring. MPoC SDK2 provides payment routing and messaging services, which may include forwarding of (encrypted) payment-related cryptographic keys and formatting of messages so they can be accepted by the payment host.

Here, MPoC SDK2 would be assessed against any relevant requirements in MPoC Domain 1, such as the Secure Channel requirements, as well as the requirements of Domain 2A (if MPoC SDK1 is an Isolating SDK). However, depending on the implementation, MPoC SDK2 may not be required to be validated to other requirements in Domain 1—if it is not handling cleartext account data or security related cryptographic keys.

Finally, in Example (4) we have an implementation where MPoC SDK1 is integrated into another SDK (MPoC SDK2) that adds non-payment features—such as reading of transport cards through the COTS-Native NFC. If this MPoC SDK2 is to be implemented in a way that allows MPoC Applications to integrate it through validation to Domain 2A, MPoC SDK2 must be listed along with the associated MPoC Product.

In this case, as long as MPoC SDK2 does not interact with cleartext assets or account data, MPoC SDK2 may be validated to Domain 2A.

MPoC Domain and Section Applicability

An MPoC Solution may involve up to three types of Entities—an MPoC Solution provider, an MPoC Software vendor, and an MPoC Service provider. The MPoC requirements applicable to each are illustrated below. An MPoC Solution that includes no other listed MPoC Products is always considered a monolithic MPoC Solution.

The MPoC standard allows for an entity to take on more than one role in an MPoC Solution. For example, an MPoC Software vendor may choose to also take the role of an MPoC Solution provider while outsourcing the A&M operation to an MPoC Service provider. In this case, the MPoC Solution may consist of an MPoC Application that exposes payment APIs for other applications to call, rather than an MPoC SDK to be integrated into new MPoC Applications (although this would also be possible).

In another example, an MPoC Software vendor may take the role of an MPoC Service provider themselves, operating their back-end software for different MPoC Solution providers. Other possible combinations also exist. However, in all cases, the requirements that apply to that role will apply to any entity taking on that role.

The MPoC standard allows for modular evaluation and certification of various MPoC Services and MPoC Software products. These are intended to be integrated into a listed MPoC Solution or developed entirely as a monolithic MPoC Solution that does not rely on any other listings, for merchant deployment.

A monolithic MPoC Solution does not use any other listed MPoC Products, and therefore must always use a monolithic MPoC Application, and an internally developed and operated Attestation and Monitoring system. If an entity wants to create an MPoC Solution with MPoC Applications that integrate an MPoC SDK, and/or that uses an outsourced Attestation and Monitoring Service, that MPoC Solution must be based on a listed MPoC Software Product.

Table 2: MPoC Requirements Applicability Matrix

MPoC Software Core Requirements	MPoC Software	MPoC Service *(note 8)*		MPoC Solution Integrating MPoC Software	MPoC Solution Integrating A&M Service	MPoC Solution Integrating A&M and Payment Service	Monolithic MPoC Solution
		A&M	A&M & Payment Processing				
Domain 1: **Module 1A: CORE**							
1A-1 Secure Software Requirements	✓						✓
1A-2 Random Numbers	✓						✓
1A-3 Acceptable Cryptography	✓						✓
1A-4 Key Management Design	✓						✓
1A-5 Secure Channels	✓						✓
1A-6 Third-Party APIs	✓						✓
Module 1B: MPoC SDK Software Protection							
1B-1 Software Security Mechanisms	✓						✓
1B-2 Software-Protected Cryptography	✓						✓
Module 1C: Attestation and Monitoring Software							
1C-1 Coverage	✓						✓
1C-2 Measurements/Detection	✓						✓
1C-3 Response	✓						✓
1C-4 Anti-Tampering	✓						✓
1C-5 A&M Integration Guidance	✓						✓

MPoC Software Core Requirements	MPoC Software	MPoC Service *(note 8)*		MPoC Solution Integrating MPoC Software	MPoC Solution Integrating A&M Service	MPoC Solution Integrating A&M and Payment Service	Monolithic MPoC Solution
		A&M	A&M & Payment Processing				
Module 1D: Secure Entry and Processing of Account Data							
1D-1 Account Data Entry and Encryption	✓						✓
1D-2 Use of PCI PTS POI-approved Devices	1						1
1D-3 Magnetic-Stripe Data	1						1
1D-4 COTS-native NFC Interface	2						2
1D-5 Manual Entry	1						1
Module 1E: PIN Entry on COTS Device							
1E-1 COTS-native PIN Entry	2						2
Module 1F: Offline Payment Transactions							
1F-1 Offline Payment Transactions	1						1
1F-2 Offline Monitoring	1						1
Module 1G: MPoC Software Security Guidance							
1G-1 Security Guidance	3						
Domain 2: MPoC Application Integration **Module 2A: MPoC Software Integration**							
2A-1 Secure MPoC SDK Integrate & Use	3, 10		3, 10	3	3	3	
Module 2B: MPoC Application security							
2B-1 MPoC Application Security	3		4	4	4	4	

MPoC Software Core Requirements	MPoC Software	MPoC Service *(note 8)*		MPoC Solution Integrating MPoC Software	MPoC Solution Integrating A&M Service	MPoC Solution Integrating A&M and Payment Service	Monolithic MPoC Solution
		A&M	A&M & Payment Processing				
Domain 3: Attestation and Monitoring **Module 3A: MPoC Software Implementation Guide Compliance**							
3A-1 Deploy & Cnfg of Back-end Systems		✓	✓	✓			✓
Module 3B: Attestation and Monitoring							
3B-1 Attestation and Monitoring Policy		✓	✓	✓			✓
3B-2 Monitoring		✓	✓	✓			✓
Module 3C: Operational Security							
3C-1 Operational Management		✓	✓	✓			✓
Domain 4: MPoC Software Management **Module 4A: Software Management**							
4A-1 COTS SW Distribution and Updates	5		5	✓	✓	✓	✓
4A-2 Key Management Operations	6	✓	✓	✓	✓	6	✓
4A-3 COTS Baseline and Vuln Management	✓	✓	✓	✓	7	7	✓
4A-4 Security of Back-end systems	9	✓	✓	✓	✓	✓	✓

MPoC Software Core Requirements	MPoC Software	MPoC Service *(note 8)*		MPoC Solution Integrating MPoC Software	MPoC Solution Integrating A&M Service	MPoC Solution Integrating A&M and Payment Service	Monolithic MPoC Solution
		A&M	A&M & Payment Processing				
Domain 5: MPoC Solution **Module 5A: Third-Party Management**							
5A-1 Merchant Identification and Comms			✓	✓	✓	✓	✓
5A-2 Support for Multiple Entities				✓	✓	✓	✓

Notes:

1,2 These requirements are to be assessed for MPoC Products that support these methods of account data presentment and processing.

2 An MPoC Product must implement one or both of COTS-native NFC or COTS-native PIN entry.

3 This Module is only applicable if MPoC Applications integrating an MPoC SDK are supported.

4 This Module is only applicable to MPoC Applications that integrate a non-Isolating MPoC SDK, or that are found to not correctly integrate an Isolating MPoC SDK.

5 This Section applies when an aspect of the MPoC Product is directly distributed to merchants or end users.

6 This Section applies to an MPoC Product when ongoing key management operations are performed, such as updates to software protected cryptography.

7 Only Requirement 4A-3.1 and 4A-3.2 apply to MPoC Solutions integrating an A&M service.

8 An MPoC Service must either integrate an existing MPoC Software product, or additionally meet all requirements that otherwise apply to an independent MPoC Software product.

9 Requirement 4A-4.1 applies to any MPoC Product that implements back-end or remote systems (including remote kernels).

10 Section 2A may be used to assess MPoC SDKs that integrate another Isolating SDK, if the MPoC SDK that is performing the integration does not provide additional card-payment features, or otherwise access any cleartext assets.

Example MPoC Implementations

This section provides some examples of how an MPoC Product may be realized. The examples provided are non-exhaustive, and other types of implementations—with different combinations of listed MPoC Products and MPoC Applications—may be possible.

Note: *An MPoC Solution that relies on (one or more) MPoC Service(s) needs to ensure the MPoC Applications it deploys are supported by those services.*

Monolithic MPoC Solution Examples

A monolithic MPoC Solution is defined by the fact that it does not integrate or use any other listed MPoC Products. A monolithic MPoC Solution may include or use external card readers, such as a PCI PTS POI or non-PTS approved MSR. Monolithic MPoC Solutions are assessed to all Domains of the MPoC Standard, although some Modules/Sections/requirements may not apply where these are scoped solely for the integration or assessment of different MPoC Product or account data-entry types. Refer to Section: MPoC Domain and Section Applicability above for details on the specific Modules and Sections that may apply.

A monolithic MPoC Solution may implement multiple MPoC Applications, each with different account data-entry methods and CVM support. For example, a monolithic MPoC Solution may have four MPoC Applications, split across two different operating systems. For each operating system there may be one MPoC Application that supports COTS-native NFC (with optional PIN as CVM), and one MPoC Application that supports card reading through a PCI PTS POI (with optional PIN as CVM).

Alternatively, this example MPoC Solution may support only two MPoC Applications—with a single MPoC Application on each supported OS providing for both PCI PTS POI and COTS-native NFC reading (both with optional PIN as CVM).

In all cases, a monolithic MPoC Solution will develop its own MPoC Applications and implement its own Attestation and Monitoring systems to support the MPoC Application(s) it deploys.

MPoC Solution Implementing a Single MPoC Software Product with an MPoC Service (A&M)

An MPoC Solution that integrates or uses one or more listed MPoC Products is no longer considered a monolithic MPoC Solution. In this example, an MPoC Solution integrates a single MPoC Software Product—integrating the MPoC SDK into one or more MPoC Applications—and (in this example) also relies on the use of a listed MPoC Service (A&M) provider who supports the same MPoC Software Product that includes the MPoC SDK that is used.

In this example, the MPoC SDK being integrated supports an external non-PTS approved MSR and COTS-native NFC (with optional PIN support for the contactless payment channel). The MPoC Solution is deploying two MPoC Applications - one of the MPoC Applications implements both payment acceptance channels supported by the SDK (COTS-native NFC and non-PTS approved MSR), and one implements only the COTS-native NFC (with optional PIN) functions of the MPoC SDK.

In this case, the MPoC Solution would not be assessed against Domain 1 or Domain 3, as it is not implementing its own software or attestation and monitoring systems as part of the core of the MPoC Solution (if the MPoC Solution were to implement its own attestation and monitoring systems, based on the MPoC Software Product used, Domain 3 assessment would be included). The MPoC Applications would be assessed against Domain 2, either Module 2A only (if the MPoC SDK is an Isolating SDK and is found to be correctly implemented), or both Modules 2A and 2B (if the MPoC SDK is non-Isolating). The MPoC Solution will also be assessed against the requirements of Domain 4 and 5 of the MPoC Standard.

The MPoC Software Product will have been assessed to Domain 1, and the Attestation and Monitoring Service to Domain 3, prior to listing of those MPoC Products.

Some requirements of Domain 4 may also have been assessed against the MPoC Software vendor and the MPoC Service provider, e.g., in the case the Entity manages their own software-protected cryptography implementation (with associated key management requirements). However, even if this is the case, key management requirements would remain in scope for the MPoC Solution as well (e.g., as they relate the management of PIN keys).

An MPoC Solution that integrates an MPoC Software Product may still implement monolithic MPoC Applications, but any such monolithic MPoC Application must be supported by its own Attestation and Monitoring systems (as the Attestation and Monitoring component of a listed MPoC Software Product will only support the MPoC SDK that is part of that MPoC Software Product). In such cases, the MPoC Solution will be assessed to Domain 1 and Domain 3 so that the security of the monolithic MPoC Application and attestation and monitoring systems can be validated.

MPoC Service Implementing A&M and Payment Processing

An MPoC Service may be implemented which combines into a single listing all of the aspects of an MPoC Software Product, as well as an MPoC A&M Service and payment processing functions. This implementation would be assessed against Domains 1, 3, and 4 at least—with Domain 2 included if the listing contained MPoC Applications and/or MPoC SDKs that integrate another MPoC SDK.

Such an MPoC Service may include both MPoC SDK listings as well as MPoC Application listings. The MPoC Applications listed as part of that MPoC Service may be developed by another entity.

MPoC Solution Implementing a Single MPoC Service (A&M and Payment Processing)

An MPoC Solution could integrate a listed MPoC Service providing A&M and Payment Processing (as outlined above) and be assessed only to the requirements in Domain 5, as well as the requirements of Domain 2 for any MPoC Applications listed with that MPoC Solution (that have not already been assessed and listed as part of another MPoC Product that MPoC Solution is integrating). This provides for a light-touch path to validation for "white-label" implementations, and for entities choosing to build an MPoC Solution where they are not themselves performing any of the software development or maintenance, key management, or vulnerability management.

MPoC Solution Implemented by MPoC Software Vendor

An MPoC Software vendor may choose to implement their own MPoC Solution, potentially as a completely monolithic MPoC Solution, or based on their separately listed MPoC Software Product and MPoC Service. Assessment in this example would follow the examples given above, for either a monolithic MPoC Solution, or for an MPoC Solution implementing a single MPoC Service.

MPoC Solution Implementing Multiple MPoC Software Products with Multiple MPoC Services

In another example, an MPoC Solution may integrate multiple MPoC Software Products, in conjunction with one or more associated MPoC Services. In this example, the MPoC Solution integrates two MPoC Software Products; one of these supports only COTS-native NFC entry (with optional PIN for CVM) on a specific COTS OS (designed herein as OS(a)), and the other supports external reading of account data (using a PCI PTS POI) on two COTS OS's (the previously designed OS(a) and another COTS OS designed OS(b)) in addition to COTS-native NFC reading on OS(b) (both with optional PIN CVM).

The example MPoC Solution deploys two MPoC Applications. The MPoC Application deployed on OS(b) integrates a single MPoC SDK. The MPoC Application deployed on OS(a) integrates two MPoC SDKs, one MPoC SDK to support COTS-native NFC (with optional PIN CVM) and the other MPoC SDK to support reading through the PCI PTS POI (with optional PIN CVM).

The COTS-native PIN entry is separately managed by each MPoC SDK, as the PIN (or other sensitive assets) cannot be passed outside of the MPoC SDK boundary in cleartext. Although this may lead to a different user experience for PIN entry for the MPoC Application deployed on OS(a), COTS-native PIN entry for any transaction must always be managed by the MPoC SDK that is used to read the payment card for that transaction.

In this example MPoC Solution is not assessed to Domain 1 or Domain 3, as it relies on the MPoC Software that it integrates and does not implement any monolithic MPoC Applications. Each MPoC Application is assessed to Domain 2, with the MPoC Application that targets OS(s) being assessed with respect to both of the MPoC SDKs it integrates.

Relationship between This Standard and Other PCI Standards

Various security requirements in this standard are based on elements of, or share similarities with, other PCI standards, as follows:

- COTS device, attestation, and monitoring security controls to protect the security of payment transactions on COTS devices are consistent with PCI Software-Based PIN Entry on COTS (SPoC) and PCI Contactless Payments on COTS (CPoC).
- POI devices are approved per PCI PIN Transaction Security (PTS) Point of Interaction (POI) requirements.
- HSMs in the back-end environment used for PIN and account-data decryption, and related cryptographic-key operations require validation to PCI PTS HSM or FIPS 140-2, or 140-3, Level 3 (or 4).
- Software used in the solution and software lifecycle practices are developed using best practices consistent with the PCI Software Security Framework (SSF).
- The back-end payment-processing environment is required to be PCI DSS compliant.
- The back-end PIN-processing environment is required to be PCI PIN Security compliant.
- The security requirements for back-end attestation and monitoring environment are developed from PCI DSS (where the back-end attestation and monitoring systems are sufficiently isolated from any account data processing).

Note: *This standard does not supersede the requirements of any other PCI standards (e.g., PCI Data Security Standard, PCI PIN Security Requirements), 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.*

Relationship between This Standard and PCI DSS

There is an independent relationship between this standard and PCI DSS. A back-end attestation and monitoring environment could be part of a cardholder data environment (CDE) or be completely separate from any CDE. If a back-end attestation and monitoring environment contains account data, it is subject to PCI DSS in accordance with payment brand compliance programs.

If an entity has already applied PCI DSS to protect its back-end attestation and monitoring environment as part of its CDE, the entity may be able to leverage the results of its PCI DSS assessment to meet the security requirements in this standard. For details, refer to *Appendix A*.

Relationship between This Standard and PCI PTS POI Standard

The PCI PTS POI standard supports the use of secure hardware and Point of Interaction (POI) devices within the payment ecosystem. The use of PTS POI devices in the solution to store, process, or transmit account data prevents exposure of sensitive assets on COTS devices and facilitates the principle of strong isolation of PIN and PAN. For more information about the PCI PTS POI, including applicability to different types of hardware, refer to the PCI PTS POI Program Guide at *www.pcisecuritystandards.org*.

Use of PCI PTS POI Devices

A PCI PTS POI used with an MPoC Solution must be an approved PCI PTS POI device that is listed on the PCI SSC Approved Device website with SRED functionality. This class of devices may optionally support contact magnetic-stripe reading functionality. When included with an MPoC Solution, a PCI PTS POI may be used for any method of card data presentment that it supports, including for the acceptance of contactless payment cards, as long as all account data output is encrypted. A PCI PTS POI may also be included when a COTS-native presentment method is used in place of one supported by the PCI PTS POI—e.g., a PCI PTS POI that supports contactless cards could be used with an MPoC Solution that uses COTS-native NFC acceptance instead.

Use of Non-PTS Approved MSR

When non-PTS approved MSR is supported by the solution, the tester must validate the standalone non-PTS approved MSR device against specific requirements in *Appendix F: MSR Security Requirements* that focuses on encryption of account data on the non-PTS approved MSR device.

Note: *PIN entry is not permitted for any magnetic-stripe-based transactions, regardless of the entry method used (PCI PTS POI or non-PTS approved MSR).*

Relationship between This Standard and PCI SSC Software Standards

PCI SSC supports the use of secure payment software within entities' cardholder data environments via the PCI Security Software Framework (SSF). The SSF consists of the Secure Software Standard and the Secure Software Lifecycle (Secure SLC) Standard. Software that is PCI SSC validated and listed provides assurance that the software has been developed using secure practices and has met a defined set of security requirements. Entities that develop their own software are encouraged to refer to PCI SSC's security software standards and consider the requirements therein as best practices to use in their development environments. Secure payment software implemented in a PCI DSS-compliant environment will help to minimize the potential for security breaches leading to compromises of account data and the damaging fraud resulting from these breaches.

Relationship between This Standard and PCI PIN Standard

MPoC Solutions may be used for the acceptance of cardholder PINs. The security of the key management and cryptographic processes used to handle customer PINs are covered by the PCI PIN standard, and therefore this standard requires compliance to the PCI PIN requirements for any back-end systems involved in PIN processing. Requirement 1 of the PCI PIN standard outlines a need to have all PIN acceptance devices approved to PCI PTS and does not need to be assessed as compliant for the MPoC implementation.

Relationship between This Standard and PCI SPoC Standard and PCI CPoC Standard

The MPoC standard incorporates and builds upon many of the concepts and requirements found in the PCI SPoC and PCI CPoC standards. However, the MPoC standard does not supersede or replace these other mobile standards. For details of any migration path from an existing SPoC or CPoC Solution to the listing of an MPoC Solution, refer to the MPoC Program Guide.

Security Requirements for Mobile Payments on COTS Solution

Objective-Based Approach to Requirements

The security and test requirements of this standard address known attack scenarios at the time when the standard was published. Any entity responsible for some component of an overall MPoC Solution has ongoing responsibility to proactively perform risk assessments to identify potential security flaws in transaction scenarios that were introduced by changes in technology or by the identification of new threats and vulnerabilities.

For an objective-based approach to be successful, entities are expected to possess a robust risk-management practice as an integral part of their "business-as-usual" operational process. While this approach provides the entities with the flexibility to implement security controls based on identified risk, the entity needs to be able to demonstrate how the implemented controls are supported by the results of its risk-identification and risk-management practices. Without a robust risk-management practice and evidence to support risk-based decision making, adherence to the requirements in this standard may be difficult to validate.

If security requirements do not define a specific level, rigor, or frequency for periodic or recurring activities (e.g., the maximum period in which an entity is required to release a security update to fix a known vulnerability), the entity may define the level of rigor or frequency that is appropriate. The rigor and frequency defined by the entity must be supported by documented risk assessments and the resultant risk-management decisions. The entity is expected to be able to demonstrate that its implementation provides ongoing assurance that the security controls or security activities are effective and meet all applicable requirements.

Requirement Frequency

Certain security and test requirements have been established with specific timeframes for activities that must be performed consistently via a regularly scheduled and repeatable process. The intent is that the activity is performed at an interval as close to that timeframe as possible without exceeding it. The entity has the discretion to perform an activity more often than specified (e.g., performing an activity monthly where the security requirement specifies it be performed every three months).

Table 3. Security Requirement Timeframes

Timeframes	Descriptions and Examples
Daily	Every day of the year (not only on business days).
Weekly	At least once every seven days.
Monthly	At least once every 30 to 31 days, or on the n^{th} day of the month.
Every three months ("quarterly")	At least once every 90 to 92 days, or on the n^{th} day of each third month.
Every six months	At least once every 180 to 184 days, or on the n^{th} day of each sixth month.
Every 12 months ("annually")	At least once every 365 (or 366 for leap years) days or on the same date every year.
Periodically	Frequency of occurrence is at the discretion of the entity and is documented and supported by the risk assessment. The entity must demonstrate that the frequency is appropriate for the activity to be effective and to meet the intent of the requirement.
Immediately	Without delay. In real time or near real time.
Promptly	As soon as reasonably possible.
Significant change	There are certain requirements for which performance is specified upon a significant change in the MPoC Software, back-end, or the solution. While what constitutes a significant change is highly dependent on the configuration of a given environment, supported COTS OSs and COTS devices, MPoC Software architecture, etc., each of the following activities, at a minimum, has potential impacts on the security of the solution and must be considered as a significant change in the context of related requirements: • Changes to hardware-based or software-based security control to protect cryptographic materials • Any changes to the underlying supporting infrastructure of the solution (including, but not limited to, changes to attestation, and monitoring system) • Any changes to third-party vendors/service providers or services provided that support the solution or meet security requirements on behalf of the MPoC Solution provider (e.g., identification of security vulnerabilities in an unsupported OS)

For other requirements, where the standard does not define a minimum frequency for recurring activities but instead allows for the requirement to be met "periodically," the entity is expected to define the frequency as appropriate for its business.

Requirements Structure

The security requirements defined within this standard are presented in the following format:

- **Security Objective**. Identifies the high-level security objective that the entity is required to meet. Security objectives are broadly stated to enable entities flexibility in determining the best methods to achieve the stated security objective. However, it is expected that the entity produces clear and unambiguous evidence to show that the chosen methods are appropriate, sufficient, and properly implemented to satisfy the security objective. Below the security objective, additional information has been provided to help both entities and laboratories understand the intent behind the security objective.
- **Security Requirements**. Specific security controls or activities that must be implemented by the entity to support the overarching security objective.
- **Test Requirements**. Describe the expected testing activities to be performed by the laboratory to validate whether an entity has met a particular security requirement. The test requirements are intended to provide both the entity and the laboratory with a common understanding of the assessment activities to be performed. The specific methods and items examined, and the personnel interviewed, are required to be appropriate for the security objective and associated requirements being assessed and for each entity's particular implementation.
- **Guidance**. Additional information to help entities and laboratories understand the intent of each requirement. The guidance may also include best practices that should be considered as well as examples of controls or methods that, when properly implemented, may meet the intent of the requirement. This guidance is not intended to preclude other methods that an entity may use to meet a requirement, nor does it replace or extend the requirements to which it refers.

Testing Methods

Entities are expected to produce evidence that they have satisfied the security requirements defined in this document. The test requirements for each security requirement describe the activities to be performed by the tester to demonstrate that the entity has met that security requirement. Where the tester finds it necessary to develop alternative tests, they must provide appropriate justification for their use. Test requirements typically include the following activities:

- **Examination**. The tester critically evaluates evidence. Common examples of evidence include software design and architecture documents (electronic or physical), source code, configuration, and metadata files, bug tracking data, and other output from software-development systems, and security-testing results. The choice of which evidence may be used to meet an examination requirement is deliberately left open for the tester to determine. However, it is a requirement of this standard that the source code of the MPoC Software and MPoC Application is made available for review as part of the assessment. It is not acceptable for an evaluation report to be provided where no source code was examined or used in the process of performing the testing.
  Where this standard uses the term "document," this is not required to be a formal physical document. Other types of managing information may be acceptable if they contain the required information and have the required utility of purpose.

- **Testing**. The tester evaluates the solution code or the operation of the solution software using a variety of security-testing tools and techniques. Examples of such tools and techniques include the use of automated static analysis security testing (SAST), dynamic analysis security testing (DAST), interactive application security testing (IAST), and software composition analysis (SCA) tools. Manual techniques, such as manual code reviews, penetration testing, side-channel attacks, fault injection, and memory scraping, may need to also be considered.
- **Observation**. The tester watches an action or views something in the environment. Examples of observation subjects include personnel performing tasks or processes, software or system components performing a function or responding to input, system configurations/settings, environmental conditions, and physical controls. Observation may include the performance of "tests," so that the output of those tests may be observed, potentially under changing conditions as the input is manipulated by the tester or other systems.

  An "observation" test process generally differs from a "testing" test process in that it involves some aspect of the normal operation of the system under test rather than testing of some subsystem or subfunction. For example, a process involving validation of protections against Man-in-the-Middle attacks through manipulation of a TLS connection from a functioning system would be "observation." Side-channel analysis of the cryptography implement during the TLS process would be performed as part of "testing."
- **Interview**. The tester converses with individual personnel. The purpose of interviews includes determining how an activity is performed, whether an activity is performed as defined, and whether personnel have particular knowledge or understanding of applicable policies, processes, responsibilities, and concepts.
- **Document**. The tester provides details or information in the evaluation report, which may be used in the same or subsequent testing requirements.

The test requirements provide both entities and testers with a common understanding of the validation activities to be performed. The specific items or processes to be examined or observed and personnel to be interviewed are required to be appropriate for the security requirement being validated and for each entity's structure, operations, and business practices. For example, it is expected that not every item of information will be contained in a formal document, and not every interview will be conducted in person. It is at the discretion of the tester to determine the appropriateness or adequacy of the evidence provided by the entity to support each security requirement. Where bullets are specified in a security requirement or test requirement, each bullet is expected to be tested as part of the validation.

When documenting the assessment results, the tester identifies the testing activities performed and the result of each activity. While it is expected that the tester will perform all the test requirements for each security requirement, it may also be possible for a security requirement to be validated using different or additional testing methods. In such cases, the tester is expected to document why alternative testing methods were used that differed from those identified in this document, and how those methods provided at least the same level of assurance as the documented testing methods. Where terms such as "periodic," "appropriate," and "reasonable" are used in the test requirement, it is the entity's responsibility to define and defend its decisions regarding the frequency, robustness, and maturity of the implemented controls or processes.

Security Objective and Assets

This standard sets forward various security control objectives for the purposes of protecting assets. Within the context of this standard, assets are elements of the MPoC Solution that are security sensitive or are used to provide security to other security-sensitive elements. Examples of assets include data such as account data, cardholder PINs, certificates, and cryptographic keys. Software may also be considered an asset if the correct operation of that software is required to provide security protection to other data assets.

Sensitive assets are a sub-set of the asset class that require confidentiality protections.

The specific assets used in a solution are expected to be unique to how that solution operates, and therefore a comprehensive list is required to be developed as part of the evaluation of any MPoC Solution.

The security controls required to protect payment information depend on the type of payment acceptance channels and cardholder verification methods supported by the MPoC Software. All MPoC Software is required to meet the security objective, requirements, and test requirements in the Core Module. The objective of these security requirements is to ensure the integrity of the COTS device, and to reasonably ensure that the solutions provide adequate security mechanisms, controls, and mitigations to protect the cardholder's account data and other assets such as cryptographic keys. These requirements assist with protection from unauthorized disclosure, modification, or misuse by restricting the available attack surface and making it cost prohibitive to attack.

It is recognized that an attacker may have other objectives, such as self-promotion or nation-state attack, and may expend more resources to circumvent established controls than is warranted by the direct financial rewards.

For the COTS platform components, the objective of these security requirements is to provide reasonable assurance that these components are kept up to date and have not been tampered with.

The following table provides examples of MPoC Software assets and lists the protection required. This protection may be confidentiality (C), integrity (I), and integrity with the addition of authentication (I+). This table does not purport to be an exhaustive list of all sensitive assets that may be stored or processed by an MPoC Solution. assets not identified in this table may exist and may require protection.

Table 4: Examples of MPoC Software Assets

Data Element Type	Description	Protection Type
Account data	Account data consists of cardholder data and/or sensitive authentication data.	C & I
Cardholder Verification Method (CVM)	Method used to verify the identity and intent of the cardholder performing the transaction. Examples include PIN or signature. Use of a CVM is not mandated by this standard and some transactions may use no CVM.	C & I+
Attestation Data	Information collected from the COTS device for the purposes of validating it is in an uncompromised and secure state, suitable for performing MPoC transactions.	C & I+
Contactless Kernel	Includes split contactless kernel implementations.	I+
Cryptographic Material	Cryptographic keys and related parameters (static and ephemeral) used to protect other sensitive assets such as account data, PINs, etc., as well as establish secure channel and signing attestation data. ***Note:*** *Public cryptographic keys do not need confidentiality protection.*	C & I+
MPoC SDK	A compiled software or a library distributed for use with the MPoC Application. Includes any third-party libraries or code relied upon for the operation of the MPoC SDK	I+
MPoC Software Source Code	The source code of the MPoC Software used as part of the overall MPoC Solution. Includes code that may be present on the COTS device, as well as code that may be used on the back-end systems. MPoC Software source code may be present	C & I+
Provisioning/secret identification data	Upon installation, MPoC Applications must be provisioned with secret data, to bind the MPoC Software to the COTS platform, and cryptographic keys to secure the sensitive assets they process. This assets class is not intended to include merchant ID data used for transaction processing or other non-secret data.	C & I+

Domain 1: MPoC Software Core Requirements

The security requirements in this Domain apply to the individual components and processes that make up the MPoC Software or the equivalent areas of software provided by an MPoC Solution where a separately listed MPoC Software is not used. Wherever the terms MPoC Software or COTS-based MPoC Software are used within this Domain, requirements apply equally to the equivalent areas of an MPoC Solution that does not use listed MPoC Software.

The functional requirements of the MPoC Software can be further organized into the following components assessed under this Domain:

- A software application or a library that implements payment acceptance and optional supported cardholder verification methods and/or may influence the security of payment data processing on the COTS device (the COTS-based MPoC Software). This software may be executed, in part or as a whole, on the COTS device itself or executed remotely and rendered on the COTS device through other means.
- The back-end attestation and monitoring systems that cannot be entirely (logically) accessed from the merchant environment and that must be capable of being regularly/rapidly updated to respond to new threats and to apply changes or updates to the solutions.
- An optional API provided by the COTS-based MPoC Software that allows other third-party developers to interface with the MPoC Software.
- All contactless kernels used in the MPoC Software, regardless of their implementation within the COTS-based MPoC Software, residing on the COTS device or implemented as a remote component of a contactless kernel—e.g., cloud-based.

Optional Modules within Domain 1 include additional security objectives to protect sensitive assets associated with specific payment-acceptance channels and cardholder verification methods such as following:

- **PIN Entry on COTS device.** Includes security requirements to ensure the integrity of the PIN entry process on the COTS device. These requirements apply only to MPoC Software that supports PIN CVM.
- **Offline Payment Transactions.** Includes requirements to ensure that support for Offline Payment processing is performed securely.
- **Secure Entry and Processing of Account Data.** Although not an optional Module, this includes optional Sections that provides security requirements for account data-entry methods, such as COTS-native NFC, manual account data entry, or to ensure secure pairing with a PCI PTS POI, or non-PTS-approved MSR.

It is expected that the attestation system monitoring systems will integrate both local and remote features to allow for the identification of new types of attacks, rapid response, and deployment of updated mitigations against such threats. When MPoC Software relies on COTS hardware-based security controls or features, such as SE or TEE, these must be included in the scope of evaluation and evaluated against the applicable security requirement in this Domain.

The test requirements of Domain 1 apply to the software used in monolithic MPoC Solutions, as well as software that is designed for submission and listing as a separate MPoC Software Product.

Module 1A: CORE

These requirements are applicable to the MPoC Software. This includes the software executed on the COTS device, any Trusted Applications (TA) used, and the back-end systems software (processing, monitoring, and attestation).

1A-1 Secure Software Requirements

Software is to be developed and maintained according to a defined software-security development and lifecycle process. Software developers require knowledge to address software vulnerabilities and emerging risks.

Development of secure software requires knowledge of common attack techniques and vulnerabilities. These vulnerabilities can change over time; therefore, a continuous process to inform software developers about these changes is vital. It is not sufficient to confirm that the developers have been provided with documentation, books, and/or training on secure software development. There needs to be auditable confirmation that developers have knowledge of common vulnerabilities in the language and environment in which they develop software.

To facilitate reliable and accurate payment transactions, the systems and software used as part of the payment-transaction flow must be designed, developed, and maintained in a way that protects the integrity of payment transactions and the confidentiality of all sensitive assets stored, processed, or transmitted in association with payment transactions.

Security Requirements	Test Requirements	Guidance
Objective: Vulnerabilities in software that may pose a security risk to MPoC Software and A&M back-end assets are prevented.		
1A-1.1 MPoC Software is developed by an entity that either: • Meets the requirements of the PCI Secure Software Lifecycle (SLC) standard, or • Meets the requirements of Appendix D.	**1A-1.1.a** When the software is developed by a PCI Secure SLC-approved software vendor, the tester must confirm through examination that the entity is either listed on the PCI website and it is valid at the time of evaluation—e.g., the listing has not expired—or the tester must validate the entity against the Secure Software Lifecyle requirements.	The MPoC Software needs to be developed and maintained in accordance with secure coding standards and industry best practices to reduce the risk of vulnerabilities being introduced that result from poor coding techniques. Knowledge of industry software development standards and best practices provides information on current exploits and trends. PCI SSC publishes additional guidance, from time to time, on best practice for the assessment of environments and processes. This guidance may allow for some aspects of an assessments to be performed remotely. *(continued on next page)*
	1A-1.1.b Where the software is not developed by a PCI Secure SLC-validated software vendor, the tester must confirm through examination and observation that the MPoC Software vendor meets the requirements of Appendix D.	

Security Requirements	Test Requirements	Guidance
		Security controls to protect the integrity and the confidentiality of the sensitive assets stored, processed, or transmitted by the MPoC Software are vital for any MPoC Solution. There is no one-size-fits-all method to software security. As a result, entities need flexibility to determine the software security controls and features most appropriate to address their specific business and software risks. As such, it is required that entities possess a robust risk-management practice as an integral part of their business-as-usual operational processes and be able to demonstrate how the implemented security controls are supported by the results of their risk-management practices.
1A-1.2 A public security-flaw-reporting program is implemented to encourage the finding and reporting of vulnerabilities.	**1A-1.2.a** The tester must confirm through examination that a vulnerability-reporting program exists for the system, and there is evidence of accepting and remediating security vulnerabilities found through this program.	It is required that MPoC Vendors are able to receive and process security-flaw reports regardless of their origin. Therefore, vulnerability reporting programs are required to be publicly accessible, and clearly designated for this purpose. However, it is permissible to have more specific details of the program, such as outlines of internal processes, available to non-public groups, such as direct customers. The flaw reporting program does not need to be specific to the MPoC Product and may be a wider flaw reporting program operated by the vendor. However, it must be clear that the MPoC Product is specifically in scope of this program. It is recommended that the process for reporting vulnerabilities includes methods to secure the communications so that details of any potential vulnerabilities are not exposed prior to review and patching, as required. *(continued on next page)*
	1A-1.2.b For any vulnerabilities have been reported through the security flaw reporting program the tester must confirm through examination that any such vulnerabilities are processed through the vendor risk-and-update process and patched accordingly.	
1A-1.3 A vulnerability assessment has been performed on the MPoC Software prior to initial assessment and at least once per year thereafter.	**1A-1.3.a** The tester must confirm through examination that a vulnerability assessment has been performed on the MPoC Software prior to initial assessment and at least annually thereafter. The output of the vulnerability assessment must be examined to confirm that the scope covers all aspects of the MPoC Software, including all types of platforms supported, and that vulnerabilities found during the vulnerability assessment have been remediated or are mitigated through other protections provided by the solution.	

Security Requirements	Test Requirements	Guidance
		Penetration testing and vulnerability management processes are expected to be part of the MPoC Software vendor's secure software lifecycle process. This requirement confirms the scope and efficacy of the penetration testing as it is applied to the MPoC Software specifically. Penetration tests need to be performed by suitably skilled resources and may be performed by resources internal to the MPoC Solution provider if such resources exist. When penetration testing is performed by internal resources, the people performing the testing need to be separate from those who have been involved in the development of the MPoC Software. Skills expected from the resources used for penetration testing include: • An understanding of EMV protocols and payment processing. • Skills and experience with mobile security and communications protocols. • A clear history of penetration testing experience. Results from annual penetration testing may not exist for newly developed MPoC Software products but need to be provided for any review performed after the first year of validation. However, an initial penetration testing report is required to be available prior to the listing of the MPoC Software or the MPoC Solution (for monolithic solutions). Penetration testing may be performed by the same entity that performs the MPoC evaluation; however, the MPoC evaluation itself cannot be considered a penetration test to meet this requirement. A separate testing and reporting process must be implemented for this penetration test. This may require that the target of the penetration test (such as an MPoC Software product) is provided with a test harness to facilitate the operation of the software during the penetration testing. Any vulnerabilities identified in penetration testing must be considered during attack costings.

Security Requirements	Test Requirements	Guidance
1A-1.4 The MPoC Software implements chip-based acceptance utilizing the COTS platform for the entry of at least one form of account data.	**1A-1.4.a** The tester must confirm through examination and observation that the MPoC Software implements at a minimum (both of the following): • Acceptance for at least one of either contact or contactless chip (through COTS-native interfaces, or through use of an external PCI PTS POI). • COTS-native interfaces for the input at least one of either contactless chip, or cardholder PIN.	The MPoC standard is intended for use with solutions that use COTS-native interfaces for the acceptance of chip-based payment transactions, or cardholder verification methods (such as a PIN). Solutions that rely entirely on non-COTS devices for the acceptance of account data, and do not provide for any COTS-native acceptance of account data or PIN data, are not intended for assessment under this standard. For example, solutions that support only magnetic-stripe cards, only manual PAN entry, or do not use the COTS device for any acceptance of account data, are not to be considered for validation and listing under the PCI MPoC standard. Magnetic-stripe card and manual PAN entry transactions may be supported as optional payment channels with solutions that meet this requirement to support chip-based transactions. This requirement does not imply that an MPoC Product cannot be used for capturing cardholder data for purposes other than payment processing—e.g., for validation of payment card in a transit ticketing system.
1A-1.5 An MPoC SDK does not pass cleartext sensitive assets to another MPoC SDK or an MPoC Application.	**1A-1.5.a** The tester must confirm through examination and observation that any MPoC SDKs do not pass cleartext sensitive assets to another MPoC SDK or MPoC Application.	An MPoC SDK is intended to implement all payment functions for its intended use-case. An MPoC SDK or MPoC Application may not integrate card-based payment functionality which is not considered during the MPoC evaluation. An MPoC SDK may pass sensitive assets to an integrating MPoC Application if those sensitive assets are encrypted in line with the requirements of this standard. This requirement applies to both Isolating and non-Isolating SDKs.

Security Requirements	Test Requirements	Guidance
1A-1.6 Where an MPoC SDK integrates another MPoC SDK: • The MPoC SDKs do not share payment acceptance channel resources (such as a COTS-native NFC interface, or connection to an external card reader). • The MPoC SDKs do not share sensitive assets in a way that assets collected in one MPoC SDK could be exposed in cleartext within another MPoC SDK. • The MPoC SDK to be integrated does not itself integrate any other MPoC SDK. • The MPoC SDK to be integrated is approved and listed as an Isolating SDK, meeting all relevant MPoC requirements. • The MPoC SDK to be integrated is provided with clear guidance on how it can be securely integrated into another MPoC SDK. • The integration guidance to be followed by an MPoC Application is complete, and includes all security relevant details for all of the MPoC SDKs to be integrated into the MPoC Application. • There is no negative impact on the security of either MPoC SDK.	**1A-1.6.a** The tester must confirm through examination and observation that where an MPoC SDK integrates another MPoC SDK: • The MPoC SDKs do not share payment acceptance channel resources (such as a COTS-native NFC interface, or connection to an external card reader). • The MPoC SDKs do not share cleartext sensitive assets. • The MPoC SDK to be integrated does not itself integrate any other MPoC SDK. • The SDK to be integrated is approved and listed as an Isolating SDK, meeting all relevant MPoC requirements. • The MPoC SDK to be integrated is provided with clear guidance on how it can be securely integrated into another MPoC SDK. • The integration guidance to be followed by an MPoC Application is complete, and includes all security relevant details for all of the MPoC SDKs to be integrated into the MPoC Application. • There is no negative impact on the security of either MPoC SDK.	An MPoC SDK may integrate another MPoC SDK, if the MPoC SDK being integrated does not itself integrate another MPoC SDK, and the MPoC SDKs do not share payment acceptance resources—such as a single COTS-native NFC interface—for the purposes of card-based transaction processing. An example of a potentially acceptable implementation would be where one MPoC SDK supports COTS-native NFC payments, and another MPoC SDK that integrates this COTS-native MPoC SDK supports payments through an external PTS POI device. Alternatively, an MPoC SDK may be integrated by another MPoC SDK for the purposes of rebranding or providing additional functionality that would not be considered within MPoC scope. An MPoC SDK may not be integrated by another MPoC SDK if some previously validated aspects of the security must be disabled or bypassed to permit such integration. For example, if the A&M of the MPoC SDK to be integrated must be disabled to enable the integration. Resources may be shared with another SDK if those resources are used for purposes other than card-based transaction processing. For example, MPoC SDK2 may integrate MPoC SDK1 and both utilize the COTS-native NFC if they do not both use this interface for card-based transaction processing. In such an example, MPoC SDK1 may use the COTS-native NFC for card-based transaction processing, and MPoC SDK2 may use the COTS-native NFC for reading non-PAN-based transit cards.

Security Requirements	Test Requirements	Guidance
1A-1.7 All cleartext card-based payment functionality has been included in the scope of the MPoC Software assessment.	**1A-1.7.a** The tester must confirm through examination and observation that all cleartext card-based payment functionality has been included in the scope of the MPoC Software assessment.	Although functionality may be excluded from MPoC assessment scope in this way, any code that executes in the same memory space and/or could impact the security of the in-scope MPoC functionality, must be considered during vulnerability assessments, penetration testing, and laboratory attack costing development.
1A-1.8 The COTS-based MPoC Software provides a mechanism to validate its version number.	**1A-1.8.a** The tester shall confirm that there is a mechanism for a user to validate the version number of the COTS-based MPoC Software.	It is important that the COTS-based MPoC Software is able to be validated as correct, based on the MPoC listing. This includes ensuring there are methods to validate the version of any MPoC SDKs which are integrated into a deployed MPoC Application.

1A-2 Random Numbers

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. This makes the quality of the random numbers generated by the MPoC solution vital. Random numbers are to be generated using a process that ensures sufficient entropy and lack of statistical correlation.

Any random numbers used for security purposes must be generated using a secure method, such as a DRNG seeded from a value that comes from a trusted source. A COTS device that has a TEE or SE evaluated as a random number generation source directly can be used as a source of entropy (see Requirement 1A-2.2 guidance for more details). Otherwise, a DRNG must be used, seeded from an external trusted source such as a PCI PTS POI or a back-end system such as an HSM, in addition to entropy from the COTS device itself. Combining these different entropy sources ensures that even if one is compromised it does not automatically invalidate the security of the random number generation process as a whole.

This applies to all components and parts of the MPoC Software where random numbers are required to be generated for security functions. Random numbers that are not relied upon directly for security of the account data or attestation data, such as random values used in TLS sessions where the data being transmitted is otherwise protected using application-level cryptography, are exempt from this Section.

The COTS-based MPoC Software should maintain an entropy “pool” that is updated regularly from the trusted source and other sources on the COTS platform. This pool data is sensitive and should be protected.

Security Requirements	Test Requirements	Guidance
Objective: Random numbers are sufficiently unpredictable.		
1A-2.1 Software development documentation provides details about how the MPoC Software generates secure random numbers, as required, on all deployed platforms.	**1A-2.1.a** The tester must confirm through examination that the software-development processes and practices used, with respect to the generation of random numbers, include the required information and are consistent with the tester's understanding of the MPoC Software. Information maintained for random number generation method must include at a minimum: • The generation and origin of random numbers. • The expected entropy of any seed values used. • The seeding period. • Details of any DRNG algorithms implemented.	This requirement applies to all MPoC Software regardless of where it is executed (i.e., this requirement applies to both back-end and COTS-based MPoC Software software). Random number generation often sets the security baseline upon which other security controls rely. This may include the generation of padding data for use in certificates, key bundles, and EMV flows as well as the generation of cryptographic keys. Random generator attacks by malicious users exploit weak random number implementations and have been the cause of several high-profile vulnerabilities. Therefore, the quality of the random numbers generated by the solution is vital. The information is required to cover all uses of random numbers, including, at a minimum, the entropy sources for the attestation system and key-generation processes. The information provided needs to inform the use of all random number generation methods within the software. The remaining requirements in this Section describe the testing required to validate that the implementation is produced in line with the rules of the software-development process. Entropy supplied by any random number generation system should be sufficient for the use cases to which it is applied. For example, a random number used to generate a new key should provide entropy at least equal to that key value. For details on minimum cryptographic key strengths and other items of cryptography, refer to *Appendix C*.

Security Requirements	Test Requirements	Guidance
1A-2.2 The MPoC Software uses an assessed source for the generation of random numbers where the security of assets requires the use of random numbers.	**1A-2.2.a** The tester must confirm through examination that, where the security of assets requires the use of random numbers, an assessed Random Number Generator (RNG) is used. **1A-2.2.b** The tester must confirm, through testing against industry-recognized test suites such as NIST SP800-22 or AIS-31, that any Random Number Generators (RNGs) used for security services are fit for this purpose. ***Note:*** *For configuration and use of the NIST SP800-22 STS tool, refer to Appendix F.*	This requirement applies to all MPoC Software regardless of where it is executed —i.e., this requirement applies to both back-end and COTS-based MPoC Software. Examples of situations where random numbers may be required to secure account data include the generation of cryptographic keys, padding prior to encryption of account or PIN data, or security controls such as attestation functions. The EMV Unpredictable Number (UN) used by a payment kernel is not included in the scope of this requirement. However, where a payment kernel requires entropy for other security purposes, these random numbers are in scope of this requirement. It is required that account data and attestation data are protected with cryptographic controls. Such controls often depend on the quality of random numbers to maintain the designed security levels. The Random Number Generator (RNG) used by the MPoC Software is required to be tested for fitness of purpose against industry-recognized test suites such as NIST SP800-22 or AIS 31. It should be noted that NIST SP800-22 is a statistical test and does not test for entropy. A trusted execution environment or secure element that has an existing assessment confirming its suitability for the generation of cryptographically strong random numbers may be used where present in supported platforms. Where such random number generation hardware is not present or cannot be relied upon to be present on all supported platforms, a software DRNG is required. Previous assessments that can be used to validate the suitability of a trusted execution environment/secure element include common criteria, EMVCo, and other industry standard assessments.

Security Requirements	Test Requirements	Guidance
1A-2.3 When use of a hardware based true Random Number Generator (RNG) on the COTS device cannot be assured, the COTS-based MPoC Software implements a secure DRNG based on industry standards.	**1A-2.3.a** The tester must confirm through examination that, where use of a hardware based true Random Number Generator (RNG) on the COTS device cannot be assured, the COTS-based MPoC Software implements a secure DRNG based on well-known international and industry-standard algorithms suitable for cryptography use.	DRNG algorithms used by the COTS-based MPoC Software are required to be secure algorithms and known international standards suitable for cryptography use as specified in *Appendix C: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms*. Homegrown algorithms do not provide enough assurance on the quality of the random numbers provided or the security of the algorithm. It is expected that the algorithms used by the COTS-based MPoC Software use international and industry-standard algorithms—e.g., NIST SP 800-90A.
1A-2.4 DRNGs used by the COTS-based MPoC Software are regularly (re)seeded with unpredictable values of sufficient entropy, which are protected for confidentiality and integrity.	**1A-2.4.a** The tester must confirm through examination that, for each DRNG used in the COTS-based MPoC Software: • The seeds have appropriate entropy (at least equal to the strength of any random value they are required to produce). • The seeds are derived from trusted sources. • The seeds are protected against disclosure. • The seeds are protected against tampering. • The MPoC Software implements protections against tampering of the DRNG and its seeding process. • The DRNG is seeded at least each time the MPoC Software launches, and reseeded after every 24 hours of continuous operation. • The DRNG seeding process is performed if the integrity and confidentiality of the DRNG state cannot be ensured, such as upon launch or return from a halted state. • In all cases, the DRNG is reseeded after 24 hours has elapsed since the last seeding process. *(continued on next page)*	A DRNG is by definition deterministic—given the same input, it will produce the same output. Therefore, it is important that the input to a DRNG—the "seed"—is sufficiently unpredictable. To ensure that a compromise of a DRNG state during operation does not result in the compromise of all further values output from that DRNG, the DRNG is required to be regularly reseeded. Random seed values transmitted from external systems may be protected using the secure channel implemented for that connection. For the purposes of this requirement, "sufficient entropy" is considered to be at least the same number of bits as the effective number of bits of the largest key used.

Security Requirements	Test Requirements	Guidance
	Note: *This requirement only applies to systems using a DRNG assessed under Requirement 1A-2.3.* ***Note:*** *Appendix C outlines the requirements for minimum entropy of cryptographic keys.*	
1A-2.5 The DRNG used by the COTS-based MPoC Software uses more than one source of entropy to obtain its seed. Entropy sources include at least one external trusted source, as well as at least one trusted source from the COTS device.	**1A-2.5.a** The tester must confirm through examination that the DRNG seeding process includes random data from a trusted external source, such as a PCI PTS SCRP or back-end HSM, in addition to at least one trusted source from the COTS device. ***Note:*** *This requirement only applies to systems using a DRNG assessed under Requirement 1A-2.3.*	The DRNG used by the COTS-based MPoC Software can use different sources of entropy to obtain its seed—e.g., one HSM and the COTS platform Random Number Generator (RNG). This increases the effort needed to compromise the DRNG used by the COTS-based MPoC Software. A trusted source of entropy is one where the entropy output has been validated through testing and there is a reasonable assurance that this testing is valid for all COTS platforms in the baseline. Where testing of the COTS platform RNG is not possible or previous testing is not available, at least two sources of entropy sourced on the COTS device must be used in addition to the external source. This may include an on-device RNG along with other potential sources of randomness such as sensor or network timing inputs. The back-end random data can be obtained from the back-end HSM or other suitable source of entropy. The random data taken from the COTS device can be obtained from one of the COTS platform Random Number Generator (RNG) sources (OS, TEE, SE) or from another method of collecting unpredictable data. By using these two sets of (re)seed data from different systems (COTS device, and external) it provides confidence that the entropy sources used are independent. Where HSMs are used, the compliance and testing requirements for these are covered in the operational Domains of this standard, in requirements 4A-2.x. *(continued on next page)*

Security Requirements	Test Requirements	Guidance
		Requirement 4A-2.2 specifies that HSMs used in the back-end systems are required to be compliant to FIPS140-2 level 3 (or above) or PCI HSM. Devices approved to the SCRP approval class have been validated to provide functions to output entropy to attached devices. Where another POI approval class is used to provide external entropy, validation of the entropy provided is required. The need for an external entropy source does not apply if the COTS platform provides a suitable hardware RNG, such as through an SE or TEE, which is used by the COTS-based MPoC Software. Note that this applies only if a hardware RNG is provided, not to DRNGs, which are the focus of this requirement.

Security Requirements	Test Requirements	Guidance
1A-2.6 The reseeding method used by the COTS-based MPoC Software ensures sufficient entropy is maintained.	**1A-2.6.a** The tester must confirm through examination that the method used to reseed the DRNG ensures sufficient entropy is maintained.	When reseeding the DRNG, the new seed is required to add to the entropy of the DRNG and not be used to re-instantiate the DRNG to a previous or known state. This helps to mitigate the risk that a compromised future seed can be used to completely determine the output of the DRNG. Adding to the entropy of the DRNG, rather than implanting another seeding process to restart the DRNG, ensures that any attacker who has compromised the current entropy values is not able to determine the output of the DRNG unless they have captured all entropy seeding values. This mitigates attacks against externally supplied entropy, such as that supplied by a PCI PTS SCRP or HSM. Reseeding methods which combine the collected entropy inputs into a single seed for the DRNG maintains the entropy of each seed, such as by XORing each seed value into a single entropy pool, are examples of meeting this requirement. Some implementations may prevent the reseeding of the platform Random Number Generator (RNG). In such cases, either a separate Random Number Generator (RNG) will need to be used or methods other than reseeding will be required to ensure sufficient entropy for the platform Random Number Generator (RNG). Reseeding is not required for systems that use a hardware- based true Random Number Generator (RNG), such as secure elements or trusted execution environment assessed to provide such functions. For the purposes of this requirement, “sufficient entropy” is considered to be at least the same number of bits as the effective number of bits of the largest key used.
	1A-2.6.b The tester must confirm through examination that any entropy pool maintained for the DRNG implements methods to protect the integrity and confidentiality of that pool. ***Note:*** *This requirement only applies to systems using a DRNG assessed under Requirement 1A-2.3.*	

1A-3 Acceptable Cryptography

Cryptography is an important factor to ensure confidentiality and integrity of data and processes that support the MPoC Solution. Therefore, it is important that only industry-recognized standard cryptographic algorithms and modes of operation be the basis for any security services used in the MPoC Solution.

Sensitive assets are required be encrypted on the COTS device for transporting to other components of the MPoC Solution using cryptographic algorithms and modes of operation known to provide suitable levels of security.

All cryptographic keys are required to be used for a single specific purpose. For example, a key used to encrypt account data is not permitted to also be used to protect the integrity of the tamper-detection data.

This requirement does not apply to cryptographic methods applied by the PCI PTS POI for onward processing by the payment processing environment (e.g., PIN block translation functions performed by the PCI PTS SCRP, which have been previously assessed to PCI PTS POI standard).

Security Requirements	Test Requirements	Guidance
Objective: Industry-standard and accepted cryptography is used to protect sensitive assets.		
1A-3.1 Software-development documentation provides details about acceptable cryptographic processes and operations to be used for security services.	**1A-3.1.a** The tester must confirm through examination that the required information is present and matches the design of the solution. Documentation must include, but not be limited to, the following: • The cryptographic algorithms and key sizes that must be used for security services. These must be compliant with *Appendix C Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms* when used for security-sensitive service. • Key-generation or key-agreement processes. • Description of cryptographic key protection mechanisms. • Key derivation functions used, including any key check value, check values, or other derivation functions. • Modes of operation.	Information that identifies cryptographic operations used in the solution helps ensure that these controls and their use are appropriately understood prior to testing. It also helps to identify areas where cryptography may increase the solution's security protection. This software development documentation provides guidance and outlines the necessary controls and features for the development of the solution. Therefore, references to cryptographic algorithms or key lengths need to be in line with other testing requirements (e.g., minimum acceptable algorithm types and key lengths). The remaining requirements in this Section describe the testing required to validate that the implementation is produced in line with the rules of the software-development process.

Security Requirements	Test Requirements	Guidance
1A-3.2 All cryptographic processes, including hash functions, used to provide security to the solution adhere to *Appendix C Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms*.	**1A-3.2.a** The tester must confirm through examination that the cryptographic algorithms and the key sizes comply with *Appendix C Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms*.	To withstand attacks, the solution is required to use the most robust and current encryption algorithms and key sizes. Legacy algorithms may have known weaknesses and provide security levels that are unsuitable given current and projected computing power. Use of recognized cryptographic methods ensures that the solution adheres to industry-tested and accepted algorithms and appropriate key lengths that deliver effective key strength and proper key management practices. Proprietary or "home-grown" algorithms do not provide this assurance and are not permitted. Hash functions are used to provide integrity and support authenticity controls over data. They may be used on their own or in combination with other cryptographic controls. RSA 2048 bit may be used to load AES keys (128 bit to 256 bit) into the COTS-based MPoC Software, if use of larger RSA key sizes is prevented by the COTS platform. Loading and use of additional keys must meet minimums set for key strength equivalency within Appendix C or require deletion and reloading of the MPoC Application.

Security Requirements	Test Requirements	Guidance
1A-3.3 Public keys used by the MPoC Software are protected for integrity and authenticity and are authenticated before they are relied upon for providing security services.	**1A-3.3.a** The tester must confirm through examination and observation that the public keys used in the MPoC Software are protected for authenticity and integrity. **1A-3.3.b** The tester must confirm through examination and observation that the public keys used in the MPoC Software are authenticated before they are relied upon for providing security services.	Certificates are often used to exchange public keys. Verification of certificate signatures can be used to authenticate the public key and meta data. This is normally guaranteed by verifying the complete certificate chain up to the certificate authority. However, in the case of MPoC Software, some certificates may be signed by the MPoC Solution provider and not by an established CA. These certificates can be authenticated by using application package signature as the root of trust protecting the application-embedded certificate(s) against tampering. For certificates used by the MPoC Software that are signed by the MPoC Solution provider, a self-signed certificate embedded in the signed MPoC Software package may be used to validate other certificates. Requirements for the security of signing operations performed by the MPoC Solution provider can be found in Domain 4 of this standard.
1A-3.4 Each key has a single unique purpose, and no keys are used for multiple purposes.	**1A-3.4.a** The tester must confirm through examination that the cryptographic keys used in the MPoC Software are not used for multiple purposes, specifically noting the findings for the PIN and (other) account data keys.	The MPoC Software is required to prevent the use of a single key for more than one purpose —e.g., signing and encrypting data, or using a key encrypting key to encrypt PANs. This helps to reduce the impact of the compromise of any one key. The disclosure of a secret or private key needs to be prevented from compromising data not intended to be protected by that key—e.g., so the compromise of a key that protects card data cannot expose PIN data. Keys used in industry-standard protocols, such as TLS, are not included in this requirement. A key that exists higher in the hierarchy can be used to generate or derive multiple keys, each with its own purpose, if the key derivation method used is secure. This is assessed in the next requirement.

Security Requirements	Test Requirements	Guidance
1A-3.5 Key derivation and key check functions are implemented securely.	**1A-3.5.a** The tester must confirm through examination that the derivation and key check functions present in the MPoC Software are one-way functions and do not expose information about the keys used in the derivation or check process.	Derivation and key check functions need to be selected such that an attacker cannot gain information of the key used as part of the derivation or check process (the "derivation key" or "original key") by observing the derived value, or a set of them. Examples of derivation functions that do not reveal the derivation key are encryption and one-way functions such as CMAC.
	1A-3.5.b The tester must confirm through examination that it is not possible to calculate the derivation or key check function output without prior knowledge of the derivation or key check material, including the cryptographic key used.	

1A-4 Key Management Design

Secure key management is critical to the security of cryptographic systems. It is a fundamental factor for ensuring the confidentiality and integrity of data and processes that support the MPoC Solution. Key management practices must conform to the industry-accepted practices described in this Section. Cryptographic keys are managed securely using recognized industry requirements throughout the cryptographic lifecycle including, but not limited to:

- Generation
- Distribution/conveyance
- Storage
- Established cryptoperiods
- Replacement/rotation when the cryptoperiod is reached
- Escrow/backup
- Key compromise and recovery
- Emergency procedures to destroy and replace keys
- Accountability and audit

Secret and private cryptographic keys that are relied upon for security are required to be unique per device/application, with the exception of keys protected through software-protected cryptography means, which are used to establish an initial trust anchor prior to provisioning unique keys to the MPoC Application. Shared public keys are acceptable, but methods and procedures for revoking compromised public key/private key pairs must be implemented. For additional information about public key Infrastructure (PKI), refer to X9.79-4.

Operations that involve secret or private cryptographic keys are to be performed using split knowledge. Split knowledge requires that no one person can determine any single bit of a secret or private cryptographic key. Split knowledge can be provided in the following ways:

- Storing keys on secure cryptographic devices (SCD) approved by FIPS140-2 Level 3 (or equivalent in FIPS 140-3) or PCI PTS-HSM that will not output the cleartext key.
- Two or more full-length components during key loading.
- An M-of-N secret-sharing scheme.

The requirements for key management and cryptography apply to all methods and protocols used and relied upon for meeting MPoC requirements. Additional protocols or methods applied onto an already compliant system do not necessarily need to meet these requirements.

Security Requirements	Test Requirements	Guidance
Objective: Cryptographic keys that protect the MPoC Software and the sensitive assets are securely managed.		
1A-4.1 An inventory of all keys used by the MPoC Software is maintained.	**1A-4.1.a** The tester must confirm through examination that the information provided matches their understanding of the MPoC Solution and contains the required information for all keys used in the MPoC Software. For each key, the information must be provided containing at a minimum: • ID or name of the key • Function/purpose • Uniqueness (e.g., per transaction, device, solution, etc.) • Algorithm and key size • Cryptoperiod (key lifetime) • Key-generation location (Name of server, application, database, device, etc.) • Key-generation method (SCD, PCI PTS POI, OS, SE, TEE, software, etc.) • Key usage location (name of server, application, database, device, etc. For symmetric keys this is at least 2 locations.) • Key loading (where relevant, how is the key loaded?) • Confidentiality protection during transport • Confidentiality protection during storage • Integrity protection during transport • Integrity protection during storage • Removal/destruction	A good key management process, whether manual or automated, is based on industry standards and addresses all elements of the key lifecycle that include: • Distribution/conveyance • Storage • Established crypto periods • Replacement/rotation when the cryptoperiod is reached • Escrow/backup • Key compromise and recovery • Emergency procedures to destroy and replace keys • Accountability and audit For example, key generation is required to conform to industry-recognized procedures that ensure the confidentiality of the underlying key. Secret and private cryptographic keys need to be distributed securely, never in the clear, and only to designated custodians or recipients. Procedures for distribution apply both within the entity and outside it. Secret and private keys are required to be encrypted with a strong key-encrypting key that is stored separately, stored within an SCD (such as an HSM), or stored as at least two full-length key components or key shares in accordance with an industry-accepted method. *(continued on next page)*

Security Requirements	Test Requirements	Guidance
	1A-4.1.b The tester must document a separate key table that outlines all cryptographic keys used in the MPoC Software for the security of the solution.	A cryptoperiod needs to be identified for each key based on a risk assessment, and keys are required to be changed when this period is reached. Additionally, keys are required to be immediately prevented from use. Compromised keys should be destroyed and replaced promptly upon confirmation of a compromise. Secure key management practices include: • Minimizing access to keys to the fewest number of custodians necessary. • Enforcing split knowledge and dual control for activities involving cleartext keys or key components. • Defining roles and responsibilities for Key Custodians and Key Managers. The key inventory must include any public keys or certificates used by the solution. Where possible, certificates should be maintained in a standard format such as X.509. When certificates are used or stored in other formats, information about the certificate use and context may need to be stored in another format. For example, a specific protocol or implementation may use public keys in a format referred to as a certificate, which does not necessarily provide an "Issued to" field. When this is the case, the data may be intrinsic to the implementation or may be stored within the certificate inventory itself.
1A-4.2 Secret or private keys are imported to the MPoC Software in a form that protects their confidentiality and integrity and does not solely rely on the protections provided by any secure channel(s) being used.	**1A-4.2.a** The tester must confirm through examination and observation that secret and private keys are imported or injected into the MPoC Software only in a way that protects their confidentiality and integrity.	Requirement 4A-2.6 covers the operational aspects of key management, and notes that key blocks are one way of protecting the confidentiality and integrity of cryptographic keys. When the MPoC Software requires secret or private cryptographic keys to be imported, these need to be protected. *(continued on next page)*
	1A-4.2.b The tester must confirm through examination and observation that the confidentiality, integrity, and authenticity protections do not solely rely on the use of a secure channel.	

Security Requirements	Test Requirements	Guidance
	1A-4.2.c The tester must confirm through examination and observation that keys used to encrypt other keys for transport are not also used to secure keys during storage.	It is not sufficient to send such keys protected only using secure channel protection, such as TLS. The keys are required to have their confidentiality separately protected, such as through use of a key encryption key dedicated for that purpose. Keys should be created, and securely maintained, within the environment where they are used—e.g., hardware-backed keystore, software-protected cryptography, secure element, etc. Alternatively, cryptographic keys may be securely imported in an encrypted form, such that keys are not exposed outside of the environment where they were generated. Use of the same cryptographic keys for transport and storage can cause additional risk of exposure to the operational keys being transported or stored. Entering secret or private cryptographic keys as cleartext exposes the value of that key. Implementations need to ensure that the entry of secret or private cryptographic keys does not reduce the security of those keys. Implementations may export a secret or private key from an SCD directly into the MPoC Software, so that no one "sees" the key value, as long as the key is sufficiently protected after export—e.g., through the use of white-box cryptography for keys stored in the COTS-based MPoC Software, or storage in a hardware key store or HSM. Alternatively, an implementation may import working keys encrypted under another (transport) key. Alternatively, keys may be input using processes for "dual control and split knowledge," which provide for the entry of key components using multiple key custodians. This requirement covers only the support for secure key importing. Operational aspects are considered under the operational requirements of this standard.

Security Requirements	Test Requirements	Guidance
1A-4.3 Secret or private keys embedded into COTS-based MPoC Software implement software protection methods and are not exposed in cleartext.	**1A-4.3.a** The tester must confirm through examination and observation that if secret and private keys are embedded in the COTS-based MPoC Software, they are in a form that is protected with software-based protection measures such as software-protected cryptography.	When secret or private keys are embedded in the MPoC Software, they need to be protected using software methods such as software-protected cryptography. Embedding of secret or private cryptographic keys in aspects of the MPoC Software other than the COTS-based MPoC Software is not permitted.
	1A-4.3.b The tester must confirm through examination that any software protection measures used are compliant to the 1B-2.x requirements of this standard.	
	1A-4.3.c The tester must confirm through examination that secret or private cryptographic keys are not embedded in any other aspects of the MPoC Software, other than the COTS-based MPoC Software.	
1A-4.4 Certificates that exist on the COTS device as part of the COTS OS are considered in scope if used for security purposes.	**1A-4.4.a** The tester must confirm through examination that any certificates used by the MPoC Solution, which are part of the COTS OS, are included into the scope of the assessment.	Certificates that exist on the COTS device as part of the COTS OS may be considered out of scope if the solution is not using these certificates. Where certificates or public keys are used, they are required to meet the relevant requirements for key strength and protection.
1A-4.5 Cryptographic keys are established using a process that ensures the entropy and confidentiality of the key.	**1A-4.5.a** The tester must confirm through examination and observation that all cryptographic keys established by the MPoC Software use processes that ensure the entropy input to each key is at least equal to the effective strength of that key.	Cryptographic keys need to be established using processes that ensure their strength and confidentiality. This may include use of an approved DRNG, remote key injection, or a secure key agreement protocol. Output of cleartext secret or private cryptographic keys exposes those keys to potential compromise. For requirements outlining the effective strength of cryptographic keys, refer to *Appendix C*.
	1A-4.5.b The tester must confirm through examination, observation, and interview (where appropriate) that all cryptographic key generation processes are designed and implemented in a way that protects the confidentiality of the cryptographic keys.	

Security Requirements	Test Requirements	Guidance
1A-4.6 The MPoC Software supports the use of HSMs for storage and operation of secret and private cryptographic keys in the back-end environments. The MPoC Software design ensures that cryptographic keys used for PIN-related security functions, and all keys which are not unique per session, will never be exposed outside of a HSM in cleartext.	**1A-4.6.a** The tester must confirm through examination and observation that the MPoC Software is created to support the use of HSMs for storage and operation of cryptographic keys in the back-end environments.	Operational key management controls in *Domain 4* require the use of HSMs to secure cryptographic keys used in the MPoC Solution. To ensure that any separately listed MPoC Software product does not prevent compliance to later operational requirements, it is important that the software is created to support HSM use. Cryptographic keys which are used to secure data that may be exposed in back-end environments, such as A&M data or PAN data, may be operated outside of a HSM if the keys are unique per session. This requirement applies to secret and private keys which are used for securing sensitive assets, including other cryptographic keys, in the back-end environment. The implementation and operation of HSMs is assessed in Domain 4.
	1A-4.6.b The tester must confirm through examination and observation that the MPoC Software design ensures that any cryptographic keys used for PIN-related security functions, and all keys which are not unique per session and forward secret, will never be exposed outside of a HSM in cleartext in the back-end environment. ***Note:*** *A "session" is defined as a single transaction for any PAN-related cryptographic keys, and no more than a 24-hour period for any A&M related cryptographic keys.*	
	1A-4.6.c The tester must confirm through examination and observation that any cryptographic keys which are exposed outside of a HSM in the back-end environment are unique per session, implement forward secrecy, and are not related to PIN security functions.	
1A-4.7 The MPoC Software implements methods to revoke or otherwise cease the use of compromised cryptographic keys or certificates.	**1A-4.7.a** The tester must confirm through examination and observation that the MPoC Software is able to revoke or otherwise cease the use of cryptographic keys or certificates that are suspected of being compromised.	The secrecy and security of cryptographic keys, including the private keys associated with public key certificates, is the primary protection provided by cryptographic systems. If a key or certificate is suspected of being compromised, it is therefore vital to ensure further use of that key is prevented. Protections of this type can be implemented through methods such as certificate revocation lists (CRLs) or through explicit replacement of suspected keys / certificates.

Security Requirements	Test Requirements	Guidance
1A-4.8 Cryptographic keys are not protected with a key of lesser strength.	**1A-4.8.a** The tester must confirm through examination and observation that the MPoC Software does not allow for a cryptographic key to be protected by a key of lesser strength. ***Note:*** *An exception to this requirement is when an AES key may be protected by an RSA key of 2048 bits during transmission if the COTS platform prevents use of larger RSA key sizes. See Requirement 1A-3.2.*	It is common for cryptographic keys to be protected in some way by another cryptographic key—either through encryption (to provide confidentiality protections) or a digital signature or (H)MAC (to provide authenticity protections). If the key providing protections is weaker than the key it is protecting, this can become the most easily exploited path to the compromise of that key. The minimum strength of cryptographic keys used in an MPoC Product is noted in *Appendix C: Minimum Equivalent Key Sizes and Strengths for Approved Algorithms*. A key encrypting key may be used that is weaker than the key it is protecting, but the KEK is required to provide at least 128 bits of strength, and this is then assumed as the maximum strength of the cryptographic protection for any subordinate keys.
1A-4.9 Secret or private cryptographic keys related to account data protection are never stored as cleartext in non-volatile storage within the REE of the COTS device.	**1A-4.9.a** The tester must confirm through examination and observation that any secret or private keys used for encryption of account data are not stored as cleartext in non-volatile storage within the REE of the COTS device.	Account data-related keys are not permitted to be stored in cleartext on the COTS platform. This includes any keys used to derive the unique per transaction key that directly encrypts the account data. Use of a hardware-backed keystore to maintain the account data-related keys may assist in providing security over key storage to meet this requirement. This is assessed as part of the key management requirements. Encryption needs to use suitable cryptography as assessed in Section 1A-3 and use suitable key management as assessed in Section 1A-4.

Security Requirements	Test Requirements	Guidance
1A-4.10 Cryptographic keys used to encrypt account data on the COTS device are unique per installation of COTS-based MPoC Software.	**1A-4.10.a** The tester must confirm through examination and observation that any cryptographic keys used to encrypt account data on the COTS device are unique per installation of COTS-based MPoC Software.	Where secret or private cryptographic keys are shared across different systems, the risk of compromise for those keys increases. To ensure that the compromise of any one MPoC Application does not affect the security of any other MPoC Application, the cryptographic keys used to encrypt account data must be unique to each instance of an installed COTS-based MPoC Software (not just unique to a particular MPoC Application version). This includes public keys, if account data is encrypted with those keys, to reduce the impact of any (potential) exposure of private keys in the back-end environment. Public keys not used for account data encryption, such as keys used as part of an initial provisioning process, are not included in scope of this requirement. A common public key may be implemented for manual PAN entry, when used for the purposes of technical fallback. Where common public keys are used for technical fallback, implementations are required to comply with all other key management requirements including storage and operation of any associated private keys within a HSM (4A-2.2), and the ability to revoke any keys that are known to be compromised (1A-4.7).

Security Requirements	Test Requirements	Guidance
1A-4.11 Secret and private account data encryption keys exposed in the rich execution environment of the COTS device are unique per transaction and implement forward secrecy.	**1A-4.11.a** The tester must confirm through examination and observation that if any secret and private cryptographic keys related to account data encryption are exposed in the rich execution environment of the COTS device, those keys are implemented using a unique key per transaction key management method which provides forward secrecy. ***Note:*** *This requirement does not apply to keys used only once per application install—e.g., during initial provisioning—or keys which are used to generate future keys (but cannot be used to derive historic keys).*	Although cryptographic keys related to account data security are required to never be stored in cleartext within the rich execution environment, these keys may be exposed in the rich execution environment as cleartext during cryptographic operations. Where this occurs, it is important that the increased risk of exposure of these keys is mitigated by ensuring that the keys are unique per transaction and implement perfect forward secrecy. There should not be sufficient information left within the COTS-based MPoC Software or COTS platform to decrypt the account data after it is encrypted or to reconstruct the keys used to encrypt the data. This requirement does not apply to keys which are used only once during the initial provisioning phase, or to keys which can only be used for generating future keys (not historic keys). At all times, any keys stored in the rich execution environment must be stored encrypted.
1A-4.12 The disclosure of a secret or private key used for account data encryption does not leak any sensitive information of the key values for past keys.	**1A-4.12.a** The tester must confirm through examination that the process used for generation of account data encryption keys is secure, and that the disclosure of a key used for account data encryption does not leak any information of past keys.	The process used to generate and distribute account data encryption keys is not permitted to expose or increase the chance of compromise for any prior keys. For example, any derivation process needs to be one way. Use of DUKPT key management may be sufficient to meet this requirement, depending upon the implementation.

1A-5 Secure Channels

A secure channel is a communications connection that protects the data and assets communicated across it. Secure channels can be provided by physical means using tamper-responsive hardware or by logical means using transmission security protocols such as TLS, or application layer cryptographic methods. Secure channels are vital to protect all communications between the various MPoC components including (where present):

- Between the COTS-based MPoC Software and external card reader(s)
- Between the COTS-based MPoC Software and back-end environments

If internal systems such as a TEE or SE are used, and these provide for the use of a secure channel to protect communications between itself and the REE, then these protections must be used.

It may be acceptable for the COTS-based MPoC Software that is integrating an MPoC SDK to manage the configuration of a secure channel, or implement the secure channel in its entirety, as long as the MPoC SDK used is designed to allow for this and has specific guidance on the secure configuration and implementation of the secure channel. This guidance must be included in the MPoC SDK evaluation, and adherence to this guidance by the COTS-based MPoC Software integrating the MPoC SDK is to be validated by the MPoC Laboratory during the integration testing (which will include validation to this Section if the COTS-based MPoC Software is involved in the implementation of the secure channel).

The secure channel requirements are not required to be implemented at the transport layer. Implementations other than TLS may be acceptable if they meet the requirements of this Section.

In such cases only configuration setting by the MPoC Application is permitted (such as providing endpoint addresses and/or certificates). It is not permissible for the MPoC Application to disable any required secure channel or configure the secure channel to accept insecure cipher-suites or protocol versions.

Security Requirements	Test Requirements	Guidance
Objective: Connections between separate elements of the MPoC Solution are protected and authenticated.		
1A-5.1 Secure channels established by the MPoC Software are documented.	**1A-5.1.a** The tester must confirm through examination that the vendor secure channel information matches the design of the solution. The information must include at a minimum: • The endpoints of each secure channel. • The root of trust used for each secure channel. • Cryptography supported by each secure channel. • How the channel is established and how mutual authentication is guaranteed.	A secure design of the communication channels assists with protecting assets during transit. For a resilient design, it is necessary to have a clear understanding of how the secure channel is established and the root of trust relied upon for that secure channel. The COTS-based MPoC Software will often establish more than one secure channel for its operation. It is not necessary that a single root of trust is used for all secure channels established. Mutual authentication is not required upon first execution of the COTS-based MPoC Software, as it is expected that no instance-unique cryptographic keys will have been provisioned at this stage. However, mutual authentication is required for secure channels once the initial provisioning/personalization process has completed, and prior to the performance of any payment transactions.
1A-5.2 Connections between different physical elements of the implementation are secured through use of a secure channel.	**1A-5.2.a** The tester must confirm through examination that communication channels between different physical elements are protected using secure channels where possible. The tester must document how the secure channel is established, how the root of trust is defined and implemented, and if the secure channel is sufficient to protect the confidentiality and authenticity of the connection. When security methods rely on properties of the COTS platform instead of a secure channel, the tester must document these methods and confirm that they are suitable for use and that they are present on the platforms being used.	When elements are physically separate, a secure channel is required to protect the communications between these elements. Such secure channels need to ensure data confidentiality and authenticity during the establishment and subsequent use of the channel. No secret or sensitive assets are permitted to be transferred prior to the establishment of the secure channel, except for any data specifically used for the secure establishment of that channel. This requirement applies to aspects of the system which are physically distinct, such as remote servers and external readers. It does not apply to internal components within a COTS device. Although it is a requirement that communications between different physical components are secured, it is not always the case that a distinct transport layer secure channel—such as a TLS connection—is implemented.

Security Requirements	Test Requirements	Guidance
1A-5.3 Secret or private cryptographic keys used to establish and maintain secure channels between the elements of the MPoC Software are unique per session except by chance.	**1A-5.3.a** The tester must confirm through examination and observation that the keys used for the secure channels are unique per session except by chance.	Logical secure channels rely on cryptographic protections. Ensuring unique keys per session helps to isolate each connection, prevent replay or relay attacks, and complicate potential compromises of the cryptographic controls.
1A-5.4 Logical secure channels implement cryptographic controls for confidentiality, integrity, and authenticity.	**1A-5.4.a** The tester must confirm through examination that any logical secure channels implement cryptography compliant with *Appendix C* of this standard.	Mutual authentication between the communicating components is required to be based on cryptography that aligns with *Appendix C: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms*.
1A-5.5 Each secure channel provides mutual authentication to uniquely identify each component prior to the exchange of sensitive assets and protect against MITM and replay attacks.	**1A-5.5.a** The tester must confirm through examination and observation that the secure channels that extend outside of a physically protected boundary perform mutual authentication before any exchange of any assets.	During the installation of the COTS-based MPoC Software, it is normally not possible to authenticate the merchant COTS device as all the installable packages are the same for the same COTS device model. Currently, however, there are no unique assets provisioned to the MPoC Software and it cannot perform transactions. It is expected that after initial download, the COTS-based MPoC Software receives the necessary data to enable its authentication by the back-end in future interactions. It is not acceptable for the COTS-based MPoC Software to "re-provision" each time it is required to re-establish a secure channel, mutual authentication is required to be implemented for all connections after the initial provisioning process. Certificate pinning may be used in part to meet this requirement and is implemented by limiting the allowed certificates that can be used as a root of trust. For example, embedding a certificate in the COTS-based MPoC Software to verify the back-end certificate instead of using the platform certificate store for this purpose is one method of certificate pinning. Cryptographic mutual authentication is not required for communication connections entirely within the physical boundary of the COTS device or within the physically secure areas of the back-end processing systems.

Security Requirements	Test Requirements	Guidance
1A-5.6 The secure channels supported by the MPoC Software prevent downgrade attacks.	**1A-5.6.a** The tester must confirm through examination, observation, and testing that the secure channels are not susceptible to downgrade attacks.	A downgrade attack may result in switching to a previous version of a protocol or a lower security setting of the same protocol, such as reducing the level of encryption applied. The solution needs to prevent the downgrade of any protection levels provided by the secure channels.
1A-5.7 Assets are encrypted and authenticated at the data level during transport, and do not solely rely on the protections provided by any secure channel being used.	**1A-5.7.a** The tester must confirm through examination and observation that assets in transit are protected using application-level encryption applied at the data level in addition to the security and encryption provided by the secure channel. ***Note:*** *This requirement does not apply to platform-based attestation, which may be protected solely by the secure channel.*	The MPoC Software assets need to be protected for confidentiality and authenticity when transmitted through a secure channel. It may be possible to strip the protocol used by the secure channel, such as TLS, or intercept the data as it passes to libraries that implement the protocols used. Therefore, encryption at the data level is needed to provide an additional layer of security that cannot be stripped easily. Examples of platform-based attestation includes implementations such as Google Play Integrity.

1A-6 Third-Party APIs

All MPoC Software will expose APIs to allow for integration of the MPoC Software into the overall MPoC Solution. Some MPoC Solutions may also provide other API interfaces to other applications or systems (e.g., where the COTS-based MPoC Software performs payment functions only and receives the amount from other applications in a request for payment processing). In all cases, it is important that the APIs provided are documented clearly and implemented securely so as not to compromise the security of the payment process.

Security Requirements	Test Requirements	Guidance
Objective: Assets that are transmitted through APIs to third-party applications are protected.		
1A-6.1 Documentation of any API exposed to third parties exists.	**1A-6.1.a** The tester must confirm through examination that information provided includes the following at a minimum: • UI display data that is communicated through third-party APIs exposed by the MPoC Software. • Guidance for users of the API about how to securely use the API and protect the UI display data on the external application.	As part of the security design of the MPoC Software, the UI display data that is communicated to external parties, including the exposed API, need to be identified. As the handling of these UI display data involves an external third-party application, information needs to be included that guides the external application to take appropriate measures to protect the assets. UI elements may include personalization graphics, names, and so forth for specific users, merchants, or financial institutions, if implemented in a way that does not impact the security of the assets managed and processed by the COTS-based MPoC Software. This requirement does not mandate that UI display functions are provided to the MPoC Application or other calling applications.
1A-6.2 APIs exposed by the MPoC Software do not introduce vulnerabilities to the MPoC Solution and provide only the defined functionality.	**1A-6.2.a** The tester must confirm through examination that any APIs exposed by the MPoC Software cannot reduce the security of the overall MPoC Solution.	The MPoC Software may expose APIs to provide for integration with other systems. Such APIs are not to reduce the security of the overall MPoC Solution when used. The data transmitted through exposed APIs needs to be limited to the minimum required and never contain cleartext account data.
	1A-6.2.b The tester must confirm through examination that any APIs implemented expose only the defined functionality to provide the intended features.	

Security Requirements	Test Requirements	Guidance
1A-6.3 APIs exposed by the MPoC Software secure assets according to their required protection type.	**1A-6.3.a** The tester must confirm through examination that the protection type of the communicated assets is maintained in any API interface.	Refer to table 3 in the introductory parts of this standard for examples of MPoC assets. If assets, such as transaction data or A&M data, are to be communicated through an API, these assets are required to maintain the protection type that is required in the overall MPoC Solution. For example, if an identifier requires confidentiality protection, it is required to be communicated with its confidentiality protected and not in cleartext. Protection may be provided using cryptographic means if the assets are to be exposed in untrusted memory or processing spaces, or through the shared memory protections provided by an integrated MPoC Software. This requirement should be considered in conjunction with Requirement 1D-1.2, which covers the encryption of account data. Non-sensitive payment initiation data, such as a payment amount, is an exception to this requirement. Assets that are already sufficiently protected through encryption by the MPoC Software are also out of scope of this requirement.
	1A-6.3.b The tester must confirm through examination that, excluding non-sensitive payment initiation data such as the amount, any assets included in any APIs are protected according to their protection type.	

Module 1B: COTS-based MPoC Software Protection

These requirements apply to the COTS-based MPoC Software. The requirements of this Module 1B do not apply to the back-end aspects of the MPoC Software.

1B-1 Software Security Mechanisms

The security mechanisms that protect the COTS-based MPoC Software from attacks, such as reverse engineering, modification, and monitoring, also help protect the MPoC Solution assets as they are entered into, processed by, and transferred from the COTS device. The security mechanisms not only have a preventive role in the MPoC Software's security; they also work as detection mechanisms where they are tied to the A&M.

Many types of security mechanisms exist, each with different strengths and protection goals. It is important that the security design of the MPoC Software considers the coverage of the security mechanisms and the interaction between them. Furthermore, it is important that the level of protection (strength) of the security mechanisms is evaluated.

The requirements of Module 1B are applicable to the COTS-based MPoC Software only. They do not apply to back-end systems or software.

Security Requirements	Test Requirements	Guidance
Objective: The COTS-based MPoC Software implements security mechanisms that protect assets and resist tampering attempts.		
1B-1.1 The security mechanisms implemented in the COTS-based MPoC Software are documented.	**1B-1.1.a** The tester must confirm through examination that the information provided is complete and consistent with the tester's understanding of the solution. This information must include the list of security mechanisms included in the COTS-based MPoC Software, with the following items for each mechanism at a minimum: • What the mechanism protects against. • The party responsible for providing and/or implementing the security mechanism (i.e., the COTS-based MPoC Software or the platform). • The location or component where the mechanism is implemented (e.g., REE, TEE, SE). • The original source and provider of the mechanism (e.g., third-party vendor, MPoC Software vendor).	The COTS-based MPoC Software is required to implement protections to mitigate attacks on the COTS-based MPoC Software and assets, both during execution and when the MPoC Software is installed but not currently being executed. Multiple individual protection methods are likely to be required; therefore, it is important that each of these is documented and provided with details about how it works and what it is designed to protect. Where the COTS-based MPoC Software relies on previous evaluation or approval of subsystems, such as through use of an approved MPoC Software product, or an evaluated secure element/trusted execution environment, then the MPoC Software vendor can reference this high-level approval rather than outline each individual security feature provided by that subsystem. Details indicating what security mechanisms the COTS-based MPoC Software implement are important so that the integrating MPoC Application can ensure these security features are properly integrated and used.

Security Requirements	Test Requirements	Guidance
1B-1.2 All assets managed by the MPoC Software are identified.	**1B-1.2.a** The tester must confirm through examination that the assets are correctly identified according to the tester's understanding of the MPoC Software. The information provided must include, but not be limited to, the following: • Assets name and description. • Components that have access to the assets in cleartext or have enough information to recover it in cleartext—e.g., encrypted assets and encryption key. • How the assets are protected according to their required protection type—e.g., confidentiality, integrity, authenticity—and lifecycle stages (generation, transmission, storage, process, and removal). • Code flows involving assets.	Assets in the context of this requirement are resources or information valuable to the stakeholder or operation of the MPoC Solution that need protection. Examples of assets include cryptographic keys, account data, information supplied or used as part of an attestation system, etc. Refer to Table 4 for examples of MPoC assets. The tester needs to use their understanding of the operation and security implementation of the MPoC Software to identify the assets managed by this software during operation. The identification of assets and a strategy to protect them during the lifecycle is crucial for a secure design of the solution.
	1B-1.2.b The tester must confirm through examination that the identified assets protections are appropriate for the type of assets.	

Security Requirements	Test Requirements	Guidance
1B-1.3 Platform based security mechanisms relied upon by the COTS-based MPoC Software to protect the assets have been evaluated.	**1B-1.3.a** The tester must confirm through examination that where security mechanisms used by the COTS-based MPoC Software rely on previous approvals or certifications that the evidence and scope of prior evaluation/certification is sufficient to ensure security to the assets they protect.	If the security mechanisms used by the COTS-based MPoC Software depend on an underlying system such as a TEE, SE, etc., there needs to be assurance on the security of the underlying system. Any prior certifications relied upon are required to be relevant to the payment acceptance and security scope of an MPoC Solution. Examples of certifications that may be acceptable, include, but are not limited to: • Common Criteria (at EAL4 with AVA_VAN 5) • Common Criteria with Global Platform TEE PP • EMVCo Chip and Global Platform • EMVCo SBMP for TEE • PCI-PTS POI, PCI HSM • FIPS 140-2/FIPS 140-3 (Level 3+) The tester needs to ensure that the scope and methods used in the prior evaluation are valid for the current use case. If insufficient coverage or testing has been performed, re-evaluation of the underlying system, in part or whole, may be required by the MPoC laboratory. This requirement does not prevent the use of security mechanisms that do not have prior evaluation, but where any prior evaluation does not exist then further assessment to confirm the security features provided is expected.
	1B-1.3.b When prior evaluation is unable to be relied upon, the tester must confirm through examination, observation, and testing, that security mechanisms used by the COTS-based MPoC Software provide the expected and required security properties.	

Security Requirements	Test Requirements	Guidance
1B-1.4 Storage locations used by the COTS-based MPoC Software, including temporary buffers and caches, containing cleartext sensitive assets are cleared immediately after use. The time where a sensitive asset is available as cleartext in memory is limited to the shortest period of time possible.	**1B-1.4.a** The tester must confirm through examination about how exposure of cleartext assets is minimized and how it is ensured that such assets are cleared from storage locations immediately after use. The removal method must not rely on language/platform mechanisms—e.g., garbage collectors. ***Note:*** *This requirement applies to whenever sensitive assets are no longer required, including after early termination of a transaction due to A&M or operational controls.*	When assets (that are required to be protected for confidentiality) are to be present in cleartext in memory, the time they are present needs to be reduced as much as possible. This reduces the time that the data is exposed as cleartext and assists in ensuring that memory dumps or reuse do not expose previously processed values. This requirement does not apply to areas of memory or COTS subsystems that the COTS-based MPoC Software does not have direct access or control over (such as a COTS keystore). Instead, the COTS-based MPoC Software is expected to minimize the exposure of cleartext keys in any such areas, meeting the requirement that the time where any sensitive asset is available in cleartext to the shortest time possible.
1B-1.5 The COTS-based MPoC Software, including all sensitive assets, is resistant to reverse engineering and covers all security-sensitive areas and sensitive assets.	**1B-1.5.a** The tester must confirm through examination, observation, and testing that the methods used to provide resistance to reverse engineering sufficiently cover the required functionality, including the following functionality at a minimum when present: • The contactless kernel code. • Code handling/interfacing with cryptography functionality. • Code that handles the sensitive assets. • Security mechanisms and A&M code.	Obfuscation may not be required if the COTS-based MPoC Software is executed in a physically separate execution environment, such as an SE or TEE, which provides physical security controls that mitigate access to the COTS-based MPoC Software code. Similarly, protection against reverse engineering may not be required for all assets. Where protections are not applied to all assets, the laboratory is expected to consider attacks exploiting this lack of protection to be considered in attack costing calculations. *(continued on next page)*

Security Requirements	Test Requirements	Guidance
	1B-1.5.b Where obfuscation is used as a security feature, the tester must confirm through examination and observation that the transformations applied by the obfuscator include the ability to: • Hide data, such as (but not necessarily limited to), function/method names, strings and other data, and asset. • Modify the code flow of the COTS-based MPoC Software.	Refer to the "Security Objective and Assets" section for details on assets, and the definition of sensitive assets. Obfuscation reduces the efficacy of common code decompilation tools. Obfuscation methods may include, but are not limited to, control-flow and data obfuscation, execution of code sections in remote/cloud environments, and symbol renaming, or protections provided by virtualized execution environments that are specifically designed to provide software-based protections to code execution flows (such as a vTEE). If the COTS-based MPoC Software is provided as a number of files (libraries), the calls and interfaces between the libraries are required to be obfuscated as well. Obfuscation is intended to complicate the reverse engineering of the software and execution process of the COTS-based MPoC Software. These protections are not required across all code but need to be implemented to protect all code that handles assets or performs security checks. These protections should increase code complexity and increase the reverse-engineering effort needed to understand the COTS-based MPoC Software code. Code-shrinking tools, on their own, do not provide sufficient reverse-engineering protection.

Security Requirements	Test Requirements	Guidance
1B-1.6 Code and data provisioned to the COTS-based MPoC Software after installation is transmitted, managed, and stored securely.	**1B-1.6.a** The tester must confirm through examination, observation, and testing that code and data provisioned to the COTS-based MPoC Software after installation is transmitted, managed, and stored securely. ***Note:*** *This requirement applies to data that is security sensitive—e.g., configuration files, WebViews, keys, etc. as well as to all code that is executed by the COTS-based MPoC Software (that is not already part of the COTS Platform)—and merchant identifiers used as part of the transaction process.*	After installation, the COTS-based MPoC Software needs to be provided with unique (configuration) data to be differentiated from other installations. During provisioning/personalization assets such as key material, configuration, and unique identifiers may be provisioned to the COTS device. Data provisioned this way is not covered by the authenticity controls of the OS store and so protections need to be provided by the MPoC Solution itself. This includes any merchant identifiers required for the processing of payments, the management of which are assessed under the requirements of Domain 5.

Security Requirements	Test Requirements	Guidance
1B-1.7 The COTS-based MPoC Software prevents the use of compromised platforms which may impact the security of sensitive assets.	**1B-1.7.a** The tester must confirm through examination and observation that the COTS-based MPoC Software implements industry best practice with regard to the detection of compromised platforms and protection of COTS-based MPoC Software assets.	In the context of this requirement, detection of compromised platforms may include detection of rooting or jailbreaking, as well as other methods that may be used to compromise the integrity and security of the execution environment (such as the use of emulator systems), where this could impact the security of sensitive assets. Some MPoC platforms provide attestation functions that can be used by applications to assess the platform integrity. These mechanisms may be stronger than the ones provided by the COTS-based MPoC Software because they have insights into the platform. However, it may be possible to bypass them under certain circumstances. Understanding the limitation of these mechanisms and their proper implementation is essential for robust software hardening. Emulators facilitate the dynamic analysis of applications. The COTS-based MPoC Software is required to implement protections to help prevent its execution on these platforms to prevent such analysis. The focus of testing for this aspect of the requirement is not to consider all possible virtualization or hardware abstraction layers as non-compliant, but how the execution of the MPoC Software on any such system may facilitate dynamic analysis, and what protections the software implements to mitigate such attacks. Protections may be applied outside of the MPoC Software itself, e.g., through the utility of the execution environment. However, any security protection measures relied upon are to be considered and confirmed through evaluation in this Section. It is not sufficient to reference execution within a protected environment and provide no further testing or validation of the security of the implementation.

Security Requirements	Test Requirements	Guidance
1B-1.8 After initial download and execution, the COTS-based MPoC Software installation is securely bound to the COTS device on which it is installed.	**1B-1.8.a** The tester must confirm through examination and observation that the COTS-based MPoC Software implements methods to bind itself to the COTS device upon initial execution. It must not be possible for the bound COTS-based MPoC Software to operate on a different device or emulator platform, even if: • Data (e.g., files) can be shared from the original COTS device to the cloned device. • The second device is under attacker control.	After the COTS-based MPoC Software is installed, it goes through a process upon first execution to uniquely bind that COTS-based MPoC Software to the specific COTS device on which it is stored. The unique (configuration) data may also hold assets related to the merchant account or other operational data. The COTS-based MPoC Software is required to implement controls to prevent the extraction of data from the COTS-based MPoC Software such that it is not possible to create a "clone" of the COTS-based MPoC Software that is indistinguishable from the original. Additionally, protections binding the COTS-based MPoC Software to the COTS platform help to mitigate "code lifting" attacks. In such attacks, the COTS-based MPoC Software (or some portion thereof) is "lifted" from the COTS device so that it can be executed on another platform, or in an emulated environment, that is under the control of the attacker. Hardware-backed keystores are often used as part of the COTS device-binding solution, as these can be implemented so that it is difficult to extract the stored keys to clone the COTS-based MPoC Software. This requirement is concerned with the technical aspects of how the COTS-based MPoC Software code is bound to the COTS platform, not with the method used to bind a specific merchant identity to the COTS-based MPoC Software.

Security Requirements	Test Requirements	Guidance
1B-1.9 The COTS-based MPoC Software is developed such that the removal of the COTS-based MPoC Software results in the deletion of all sensitive assets from the COTS device.	**1B-1.9.a** The tester must confirm through examination and observation that removal of the COTS-based MPoC Software results in the deletion of all sensitive assets from the COTS device.	Provisioned and unique (configuration) data is not permitted to remain on the COTS device after the COTS-based MPoC Software has been removed. Although applications often do not have any control over their deletion, consideration during the design of the application can ensure that data is handled in a way that it is removed from the COTS device when the application is removed. Where deletion by the COTS OS cannot be relied upon, the use of cryptographic deletion may be used. Cryptographic deletion refers to when data is encrypted with strong cryptography, and the decryption key is deleted. In such cases, even when the data remains, it may be considered removed for the purposes of this requirement.

Security Requirements	Test Requirements	Guidance
1B-1.10 The COTS-based MPoC Software does not contain or expose functionality that may compromise the security of the MPoC Product, such as a developer or debug mode.	**1B-1.10.a** The tester must confirm through examination and observation that there is no option to turn on developer or debugger mode.	Prior to the release of a software build, it is common for applications to include debug or developer features that expose data and functions that would be an unacceptable risk in production systems. Such code or features are required to be removed from the distributed software object, not just disabled, prior to distribution. However, this requirement does not imply or enforce the need for two code streams (a test stream, and a production stream). Removal of test or debug code during a build process may be an acceptable way to meet this requirement. Telemetry and data collection features may remain in a COTS-based MPoC Software for the purposes of collecting A&M data or validating the ongoing correctness of the solution. However, any remaining software features are not permitted to bypass or disable the security of the MPoC Solution or the protections it provides to the payment assets it processes. For example, functions to disable encryption of account data or customer PINs, or to use “test” cryptographic keys, are not permitted. This requirement includes any COTS-based MPoC Software code implemented outside the REE of the COTS device.
	1B-1.10.b The tester must confirm through examination and testing whether developer mode or debugging code is present or whether it was removed from the installable package.	

Security Requirements	Test Requirements	Guidance
1B-1.11 When any part of the COTS-based MPoC Software functionality is implemented outside the REE, that code is also protected against tampering and handles input data securely.	**1B-1.11.a** Through examination of the COTS-based MPoC Software implemented outside the REE, the tester must confirm that the exposed interface between this code and the REE is the minimum required to perform its task.	The COTS-based MPoC Software may implement functionality external to the REE of the COTS device using TEEs, SEs, or other separate execution environments such as external devices or servers. It is required that this non-REE code is protected against tampering, either to interfere with its intended execution process or to subvert or intercept interfaces between this code and the REE. Compliance to this requirement may be achieved through demonstration of previous evaluations, such as through EMVCo SBMP, GP, or similar schemes. Documentation needs to clearly include authenticatable evidence of such evaluation—i.e., a vendor assertion of evaluation or compliance is insufficient. The tester is expected to validate the scope of any previous testing to ensure that any gaps between the previous testing and the current implementation are noted and accommodated for in the testing. For example, many SE or TEE evaluations cover only the hardware and/or operating system aspects of those systems and may not include the applications or trustlets used in the MPoC Software.
	1B-1.11.b The tester must confirm through examination or testing that any parts of the COTS-based MPoC Software implemented outside the REE are free from exploitable vulnerabilities in their implementation, such as memory corruption bugs, type confusion bugs, state management bugs, etc.	
	1B-1.11.c The tester must confirm through examination and observation that cryptographic operations and protocols implemented outside the REE are implemented and used properly and meet the requirements of this standard.	
	1B-1.11.d The tester must confirm through examination that any parts of the COTS-based MPoC Software implemented outside the REE are authenticated prior to execution.	

Security Requirements	Test Requirements	Guidance
1B-1.12 Where an Isolating MPoC SDK is claimed, the MPoC SDK is implemented in a way that secures the memory and sensitive assets of that SDK from an integrating MPoC Application.	**1B-1.12.a** The tester must confirm through examination and observation that any isolating SDK is implemented in a way that secures the memory and sensitive assets of that SDK from any MPoC Application that integrates the SDK.	Many modern operating systems allow for a single application to launch multiple processes and provide to these processes their own virtual memory space. This new memory space can be isolated from other processes, even if those processes are owned by the same user. In cases where the MPoC Laboratory is able to validate that the COTS Platform OS's targeted by the MPoC SDK are able to provide memory isolation protection to the cleartext sensitive assets, use of separate processes may be sufficient to meet the requirements for an Isolating SDK. In assessing whether an SDK can be classified as Isolating, only attacks against the built MPoC Application need to be considered. Assessment of the on-going processes used to ensure the security of MPoC Application builds will be validated during the annual checkpoint process as part of the application vendors Secure SLC process.
1B-1.13 Payment transaction data is securely deleted from the COTS device once it has been transmitted to the payment back-end.	**1B-1.13.a** The tester must confirm through examination and observation that any transaction data is securely deleted from the COTS device once it has been transmitted to the back-end system.	Data may be stored as part of a transaction prior to transmission and may include the PAN, transaction amount, and transaction cryptograms obtained from the cardholder payment instrument. To reduce the risk posed by the storage of that data, transaction data needs to be deleted after it is no longer required. However, deletion is to be delayed until after transmission of the data to process the transaction, to prevent attacks that may attempt to "force" the deletion of data by disabling communications.

Security Requirements	Test Requirements	Guidance
1B-1.14 The COTS-based MPoC Software assets and code stored on the COTS device is protected to an attack rating of 25 points using the attack-costing framework in *Appendix B*.	**1B-1.14.a** The tester must confirm through examination, observation, and testing the anti-tampering protections for COTS-based MPoC Software assets by attempting to expose or modify sensitive COTS-based MPoC Software data stored or processed on the COTS device without detection. The tester must provide a costing of this attack based on the method outlined in *Appendix B. Attack Costing Framework*. This requirement is passed if the most feasible attack cannot be costed for less than 25 points. ***Note:*** *It is not intended that the tester attempts to extract all the security assets of the COTS-based MPoC Software, but only those assets that represent the least amount of effort for an expert attacker, or the biggest gain—e.g., PIN.*	The COTS-based MPoC Software may store data on the COTS device. This data could be abused to change the behavior of the COTS-based MPoC Software in unintended ways—e.g., change COTS-based MPoC Software configuration parameters. Protection of assets in the solution is required to be sufficient to withstand expert attackers with local access to the COTS device—e.g., a malicious merchant or MPoC terminal operator with physical access to the COTS device used to accept payments. An attack example is an expert COTS device user that is able to reverse the COTS-based MPoC Software, roots the COTS device, and attempts to retrieve cleartext account data. Possible controls that can be implemented to mitigate the attack are COTS OS-level attestation, application integrity checks, etc. The combination of controls is required to be sufficient to withstand the attacks and provide the A&M the opportunity to detect the attack. The goal of this requirement is that the tester provides assurance over the protections that are documented, and the security mechanisms interaction and practicalities of possible attacks are understood. Tests performed for this requirement are required to be executed with the monitoring system disabled to test the COTS device security mechanisms in isolation. This simulates the temporary disconnection from the A&M back-end. *(continued on next page)*
	1B-1.14.b The tester must confirm through examination, observation, and testing the anti-tampering protections for the COTS-based MPoC Software code by attempting to modify the COTS-based MPoC Software code stored on the COTS device without detection. The tester must provide a costing of this attack based on the method outlined in *Appendix B. Attack Costing Framework*. This requirement is passed if the most feasible attack cannot be costed for less than 25 points.	

Security Requirements	Test Requirements	Guidance
		Code-protection mechanisms, such as obfuscation and/or platform-based protections, do not need to be disabled or specifically bypassed prior to this testing. If platform security mechanisms such as OS-backed attestation are present in the solution, the tester is required to consider the case where these OS-security mechanisms are disabled to assess the non-platform attestation functions. Examination only may be sufficient in cases where the assets are managed by a component or system that has undergone previous evaluation to a known standard, such as a HSM compliant to the PCI HSM requirements. However, in all cases, it is expected that the tester will perform sufficient testing to validate the security of the overall MPoC Solution.

1B-2 Software-Protected Cryptography

Protection of cryptographic operations and assets has been traditionally provided by tamper-responsive hardware devices such as HSMs or tamper-resistant hardware devices such as a secure element. Although use of tamper-resistant hardware components is becoming more common in COTS devices, support for them is not universal.

Another way to protect cryptographic operations and sensitive assets is through software protections, such as software-protected cryptography, where the cryptographic functions and storage methods used to protect the cryptographic keys are obfuscated such that extraction of the sensitive assets or tracing of the execution flow of the cryptographic process is rendered computationally expensive. This includes systems such as white-box cryptography, and implementations where cryptographic operations are executed in a software-protected execution environment, such as a vTEE.

When purely software methods are used to protect cryptographic keys, such as with software-protected cryptography, the specific instance used in the COTS-based MPoC Software is to be changed periodically, where the maximum period is less than the estimated time it would take to reverse engineer the software protections. It is acceptable to have two sets of keys during the changeover period, but all old keys are to be invalidated when the new keys are installed. The idea behind the update is that the software-protected cryptography implementation is updated before the needed time to break it is reached.

Using protection mechanisms that are inherent to the platform on which the COTS-based MPoC Software runs, such as hardware-backed keystore, are another option for protecting cryptographic key storage on the COTS device. 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-based and software-based protections may also be acceptable.

There are several different methods of software protection that can be applied to a cryptographic process. This standard does not mandate, require, or endorse any particular method. However, this standard is created with the expectation that cryptographic keys used to secure the MPoC Solution are securely stored and managed. The tester is expected to review the implementation and methods used to inform the testing of the robustness of these methods, so that there is confidence they provide robust protection of cryptographic process and sensitive assets they protect.

The requirements of this Section are not intended for keys operated and stored in back-end environments.

Security Requirements	Test Requirements	Guidance
Objective: Software-protected cryptography, where used, provides sufficient protection to the cryptographic keys it protects.		
1B-2.1 Cryptography protected with software-only means is documented.	**1B-2.1.a** The tester must confirm through examination that the provided information is complete based on the tester's understanding of the COTS-based MPoC Software. The information must include at a minimum: • Which keys are secured using software-protected cryptography, in whole or in part. • What algorithms are supported or implemented by the software-protected cryptography system used. • How many instances of software-protected cryptography are present for each algorithm. • The security mechanisms present in the software-protected cryptography. • The key hierarchy of the keys protected by software-protected cryptography.	Clear information is required to outline how each cryptographic key is protected so that coverage can be confirmed to be comprehensive, and flaws can be identified easily. For this requirement, it is required that the information provided covers all cryptographic keys that would otherwise be exposed in cleartext within the memory of the REE of the COTS device were it not for the implementation of software protection methods.
1B-2.2 The software-protected cryptography implementation does not implement operations that expose, or provide access to, cleartext cryptographic keys, key components/shares, or the intermediate results of a cryptographic operation.	**1B-2.2.a** The tester must confirm through examination that the software-protected cryptography operations present in the COTS-based MPoC Software do not expose, or provide access to, cleartext cryptographic keys, or key components/shares.	Implementations of software protected cryptography are intended to protect the cryptographic keys used. Protections that apply only to some aspect of the implementation, or that implement operations that may expose the cryptographic keys, are not sufficient for use with COTS-based MPoC Software. For example, a software protected cryptographic implementation that does not sufficiently protect key related values during operation, or only provides protection when the cryptographic process is not actually being executed, are insufficient. Similarly, implementations that may provide a key export feature that allows for exposure of cleartext keys or components are not sufficient.

Security Requirements	Test Requirements	Guidance
1B-2.3 The software-protected cryptography implementation does not implement unnecessary operations.	**1B-2.3.a** The tester must confirm through examination and observation that the implementation of software protected cryptography within the COTS-based MPoC Software only implements the operations required.	Usually, cryptographic implementations support at least two operations—e.g., encryption and decryption. If the COTS-based MPoC Software needs only to encrypt data before sending it to the back-end, the decryption operation is not required to be present in the COTS-based MPoC Software, since it helps reverse the encryption operation and may pose an additional security risk.
1B-2.4 The software-protected cryptography implementation, where it is used for the protection of secret or private keys embedded into the COTS-based MPoC Software, is required to be replaced before the estimated time needed to break the implementation. Replacement must include changing of any cryptographic keys, or other assets, within the software protected cryptographic implementation.	**1B-2.4.a** The tester must document an estimated exploitation time needed for an expert attacker with physical access to the COTS device to break any software-protected cryptography implementations used to protect secret or private cryptographic keys within the COTS-based MPoC Software.	The tester may use the results from the previous attack testing on the software-based cryptography implementations, to establish a time during which that implementation must be replaced to mitigate attacks. Software-protected cryptography may implement several unique master storage keys, which can be rotated over time to reduce the elapsed time during which a single key is used. However, if an attack is mounted on a software-protected cryptography implementation, this is usually automated, and rotation of master keys is of little use. Therefore, to defeat automation, the software-protected cryptography binary is required to be changed before the time estimated to extract keys from the implementation. The update of the software-protected cryptography binary may be implemented by updating the internal mathematical algorithm, its compiled binary structure, or its security mechanisms (obfuscation, anti-emulation, anti-DFA, encoding methods, etc.). The goal of the update is to impede the progress made by an attacker on the implementation.
	1B-2.4.b The tester must confirm through examination that the update cycle for the software-protected cryptography is shorter than the time needed to perform a successful attack on that implementation.	

Security Requirements	Test Requirements	Guidance
1B-2.5 The software-protected cryptography implementation supports secure key management processes, including secure key generation where key generation is implemented.	**1B-2.5.a** The tester must confirm through examination, observation, and interview (where appropriate), that the software-protected cryptography implementation supports secure key management processes, as well as secure key generation where key generation is implemented, including: • Dual control and split knowledge of keys. • Key generation methods that ensure keys have at least the same entropy as their effective key strength.	This requirement covers the implementation and support systems for any software-protected cryptography. Operation of those systems is covered under the operational requirements of this standard. This requirement does not enforce the manual handling of cryptographic keys (as key components). Proper use of back-end HSMs may provide for dual control and split knowledge across the keys.
1B-2.6 Cryptographic keys deployed within a software-protected cryptography implementation is not used for encryption of account data or secret/private cryptographic keys during transmission.	**1B-2.6.a** The tester must confirm through examination and observation that keys deployed through the software-protected cryptographic implementation are not used to encrypt account data or other secret/private cryptographic keys during transmission.	Cryptographic keys contained within a software-protected cryptography implementation during deployment may be used to protect keys during storage on the COTS Platform, or to help secure initial provisioning. However, as the keys used in these implementations may be shared across many instances of COTS-based MPoC Software, it is not acceptable to rely on software-protected cryptography mechanisms to secure account data or other cryptographic keys during transmission outside of the COTS Platform on which the implementation resides.
1B-2.7 The software-protected cryptography prevents the extraction of partial or complete cryptographic material to an attack rating of 25 points using the attack-costing framework in *Appendix B*.	**1B-2.7.a** The tester must confirm through examination, observation, and testing that the software-protected cryptography implementation is protected against known attacks against software-based cryptography. If anti-lifting or anti-emulation mechanisms are in place, separately from the A&M systems, the tester must include the attempt to bypass these mechanisms into the costing.	The software-protected cryptography needs to implement protections to help it resist the extraction of complete or partial cryptographic material. If the software-protected cryptography has a storage or transport key used to protect all other keys, the compromise of this key implies the compromise of the protected keys. This is of special concern if the storage/transport key is shared across different MPoC Software installations. *(continued on next page)*

Security Requirements	Test Requirements	Guidance
	1B-2.7.b The tester must provide a costing of this attack based on the method outlined in *Appendix B. Attack Costing Framework*. This requirement is passed if the most feasible attack cannot be costed for less than 25 points.	The security of any software-protected cryptography implementation is reliant upon periodic updates, which is validated in a separate requirement. The intent of this requirement is that the security of the software protected cryptography implementation is tested directly. Therefore, this testing is to be performed without additional security controls such as A&M or anti-rooting. The vendor of the MPoC Software or Software Protected Cryptography implementation may need to provide a test artifact for the testing process. Costing of the attack must consider the difficulty of disabling these features, and any code-lifting or emulation that is required as part of the attack. Code lifting can allow for the extraction of the software-protected cryptography implementation so that it can be executed in an environment or context under the control of an attacker. Usually, this is done for analysis of the cryptographic implementation and can, if successful, permit fault injections, oracle attacks, etc. A successful attack need not recover the entire key—it is considered sufficient if the tester is able to demonstrate success at recovery of some subset of the key and justify that this can be scaled to full key recovery. Examples of attacks to be considered by the tester includes Side Channel Analysis (SCA) and Fault Injection (FI) attacks such as Differential Computation Analysis (DCA) and Differential Fault Analysis (DFA).

Module 1C: Attestation and Monitoring Software

Attestation is the interaction between a verifier and a prover to determine the current security state/behavior of the prover based on data provided through measurements performed by the prover. For the purposes of this document, the prover is the COTS-based MPoC Software, and the verifier may have multiple components executing locally on the COTS device and remotely in a back-end attestation and monitoring (A&M).

The attestation data may be determined in various ways, such as through a health-check interface that can be accessed by the prover. Attestation provides necessary assurance to the verifier that established and expected security controls at the prover are in an acceptable state and have not been tampered with. Organizations developing A&M components are subject to these requirements.

There are two types of attestations in MPoC Solutions, where the goal is to assess the integrity of the COTS-based MPoC Software, and where the goal is to assess the integrity of the COTS platform.

It is a requirement of this standard that the attestation system and monitoring system together form the attestation and monitoring (A&M). The attestation and monitoring (A&M) includes a back-end system that can interpret and respond to the attestation results (i.e., the attestation results are required to be monitored). The COTS-based MPoC Software can attest to the COTS platform or provide attestation results to be examined by an external system—such as the attestation and monitoring (A&M). Additionally, some level of ongoing validation of the COTS-based MPoC Software and COTS platform is required both to supplement the intermittent communication with the remote attestation and monitoring (A&M), as well as to support offline payments (where applicable). However, as the COTS-based MPoC Software may be open to compromise this software cannot attest itself at all times.

1C-1 Coverage

The attestation and monitoring (A&M) must cover the COTS platform, and the COTS-based MPoC Software.

Security Requirements	Test Requirements	Guidance
Objective: The attestation and monitoring (A&M) covers all assets and processing of the COTS-based MPoC Software throughout its lifecycle.		
1C-1.1 Documentation on the coverage of the attestation and monitoring (A&M) exists.	**1C-1.1.a** The tester must confirm through examination that the required information exists, including, but not limited to: • What checks are included in the COTS-based MPoC Software attestation and monitoring (A&M) and, if applicable, which parts of their result are checked or verified. • What is covered by the checks (e.g., code, data, application, OS, cryptographic, etc.). • When, where, and under which circumstances are these checks executed and validated. • If a security mechanism covers code (e.g., checksums), which parts of the code are covered.	The attestation and monitoring (A&M) is a vital part of the overall security of an MPoC Solution and, therefore, the coverage provided by the attestation and monitoring (A&M) parts of the MPoC Software mut be clearly outlined. This information assists in the integration of the MPoC Software with both the COTS-based MPoC Software, and any back-end systems used as part of the A&M services. In the context of this requirement, the terminology “no exploitable design flaws” is to be considered in relation to the ability to exploit these flaws during an attack on the MPoC Solution. The tester is expected to consider the exploitability of any flaws during their production of attack costings during an evaluation. MPoC requires that there are checks that are performed on-device at all times, as well as checks that may be performed only when the data is to be sent to the back-end attestation and monitoring system. It is important that the documentation indicates when and where any checks occur.
	1C-1.1.b The tester must confirm through examination that the A&M design has no exploitable design flaws, matches the tester’s understanding of the MPoC Software, and is able to detect threats against an MPoC Solution.	
1C-1.2 The A&M functionality covers the complete lifecycle of the COTS-based MPoC Software, starting from installation through to decommissioning.	**1C-1.2.a** The tester must confirm, through examination, that the A&M covers the following parts of the COTS-based MPoC Software lifecycle at a minimum: • Deployment (installation) • Provisioning (enrolment, provisioning) • Operation (application startup, transaction execution, PIN entry) • Decommissioning (removal of the COTS-based MPoC Software)	The COTS-based MPoC Software could potentially be compromised at any stage of its lifecycle. Therefore, it is necessary that the A&M covers the complete lifecycle of the software deployed to the COTS device(s). It is possible that for some lifecycle stages, the COTS-based MPoC Software is not able to execute. *(continued on next page)*

Security Requirements	Test Requirements	Guidance
		However, platform features may be leveraged to protect these stages—e.g., during the deployment and decommission of the COTS-based MPoC Software. Although applications often do not have any control over their deployment and decommissioning, consideration during the design of the application can ensure that data is handled securely during these phases. The intent of this requirement is that the A&M validates the correct operation according to the design so that deployment and decommissioning can be validated as having been performed securely.
1C-1.3 The attestation and monitoring (A&M) checks cover the entire security-sensitive COTS-based MPoC Software code and execution flows that handle assets.	**1C-1.3.a** The tester must confirm through examination that: • The COTS-based MPoC Software is protected by the attestation and monitoring (A&M) at runtime. • The A&M checks provide coverage over all COTS-based MPoC Software code, execution flow, and assets. • Some A&M validation checks are always active during execution of the COTS-based MPoC Software providing protection against COTS operating modes that may impact security.	The A&M checks are required to be implemented at different execution points and be executed at different points in time to mitigate attempts to bypass or avoid the detection features. This helps to ensure that there is not a single point of failure for the A&M checks. A&M may involve different levels of checks. Some checks may be less intensive or involve call-back checks, such as checking for the activation of developer options or enabling of accessibility features that expose entry of PINs, and therefore can provide constant protection. Other attestation and monitoring (A&M) checks may be more intrusive and be possible only at intervals to prevent unwarranted interruptions to the payment-processing flow. However, the COTS-based MPoC Software is required to have some level of continuous A&M at runtime—e.g., it is not sufficient to run all A&M checks only at launch or at intervals that may allow for Time of Check Time of Use-based attacks. Such checks can be used to detect rooting or activation of developer or other modes that may affect the security of the MPoC Software operation.

Security Requirements	Test Requirements	Guidance
1C-1.4 The A&M system attests the security of the COTS platform and COTS-based MPoC Software.	**1C-1.4.a** The tester must confirm through examination that the A&M system covers the COTS Platform and COTS-based MPoC Software and is able to attest its security state.	The A&M is required to provide assurance that the COTS-based MPoC Software is not running on a compromised COTS platform—e.g., rooted, jailbroken, etc. Although the COTS-based MPoC Software can use its own checks to perform this verification, there may be value in providing data to the A&M back-end for additional validation or updates to local checks. For example, this may include validating the following information: • If the COTS device is rooted, jailbroken, or operating in developer mode (where this may impact the security of MPoC assets). • Modifications or tampering attempt events of the COTS-based MPoC Software and COTS execution environment. • The version of the COTS OS and COTS-based MPoC Software, including detection of any rollback attempts. • Details on the status of any COTS-native interfaces implemented by the COTS-based MPoC Software for account data collection. Details on the status of any COTS-native interfaces that may be used for account data collection, which are not implemented by the COTS-based MPoC Software. Checking for some potentially vulnerable states (such as jailbreaking or rooting) may not be necessary if this cannot impact the security of the MPoC assets—for example, where the COTS-based MPoC Software does not execute in the execution environment that is rooted. Some platforms provide their own attestation functions that cover some or all components of COTS platform and hardware. However, these attestation functions are not sufficient to attest the security and correct operation of the COTS-based MPoC Software (these validation requirements are tested separately). *(continued on next page)*

Security Requirements	Test Requirements	Guidance
		The attestation policy (covered in Requirement 3C-1.1) outlines the responses and actions to be performed based on the attestation results. This may vary on a per-Platform basis—e.g., a solution where the COTS-based MPoC Software resides entirely in a TEE or SE (so no sensitive assets are passed in cleartext through the REE) may not consider a rooted or jail-broken device to be a negative security impact. Similarly, detection of a roll-back of the COTS OS may not necessarily result in immediate response from the A&M system but may instead be noted and used in correlation with other signals from that same COTS device to determine the appropriate action, in line with the Attestation and Monitoring policy. A COTS Platform may not be able to attest that a specific COTS-native interface is "secure," and in such cases it may be sufficient for the attestation functions to validate that the interface is present, in good order, and not accessed by other applications. COTS platforms may provide their own logging and monitoring of interfaces that could be used for the entry of account data. For example, an OS may log all traffic on the NFC or touch interfaces. The attestation policy and system baseline should consider and account for these COTS devices. Detection of any individual item of concern during an A&M check may not result in immediate disablement of payment acceptance. However, such detection will feed into the potential for escalated observation and analysis of that COTS device to confirm it remains a secure environment in which to continue the acceptance of payments.

1C-2 Measurements/Detection

To perform attestation, the A&M component measures the COTS platform, and COTS-based MPoC Software. Some or all of these measurements are communicated to the A&M back-end for analysis and action. The quality and ability of the attestation and monitoring (A&M) to detect potential attacks are bound by the type of measurements collected by the A&M back-end, along with the analysis performed on those measurements. A&M systems are expected to incorporate knowledge across a population of COTS devices, enhancing the detection methods.

Security Requirements	Test Requirements	Guidance
Objective: Sufficient information is collected to detect and mitigate threats to the MPoC Solution.		
1C-2.1 The information that is collected for the purposes of attestation and monitoring is documented.	**1C-2.1.a** The tester must confirm through examination that the data collected for A&M purposes includes sufficient information to attest the COTS platform, MPoC Application, and MPoC SDK (where present), and identify any attached payment-card acceptance devices.	Knowledge of the signals collected from the COTS device is needed to develop the A&M service properly and assess the quality of the A&M. This requirement includes information collected on the COTS device for A&M purposes and is not related directly to security checks (e.g., COTS device model, OS version, etc.). A&M data where the COTS device or COTS-based MPoC Software is the prover may be assignable to that COTS device based on the method of collection (i.e., the signals collected can only have come from that COTS device). A&M data transmitted to the back-end A&M is required to be uniquely assignable to the COTS device from which it originated. This may be based on the cryptographic methods used to transmit that data and may use hardware-backed features present on the COTS platform, like hardware-backed keystores. Unique assignment is required so that actions performed in response to the data collected can focus on the specific COTS device to which that data relates. This does not preclude the anonymizing of collected data to enhance future A&M processing.
	1C-2.1.b The tester must confirm through examination and observation that the A&M data is uniquely assignable to the COTS device from which it originates.	

Security Requirements	Test Requirements	Guidance
1C-2.2 The A&M data reflects the current state of the COTS-based MPoC Software, COTS platform, and peripheral devices, in addition to any security relevant changes or measurements that have occurred since the last communication to the A&M back-end.	**1C-2.2.a** The tester must confirm through examination and observation that the A&M data sent to the A&M back-end reflects the current state of the COTS-based MPoC Software, COTS platform, and peripheral devices, in addition to any security relevant changes or measurements that have occurred since the last communication to the A&M back-end.	The measurements sent to the A&M back-end are required to reflect the current state of the solution. Stale measurements may provide a false sense of security or be used by an attacker to replace more current data. However, an attacker may attempt to perform attacks during the window of time between communications to the A&M back-end, so that their attacks go undetected. Therefore, in addition to ensuring the most current data, it is important that any security relevant A&M data is also transmitted to the back-end when communication is performed.
	1C-2.2.b The tester must confirm through examination and observation that any cached or historic A&M data is maintained in a manner that protects its integrity and authenticity.	
	1C-2.2.c The tester must confirm through examination and observation that any cached or historic A&M data is unable to be used to replace or subvert the most recent and accurate data reflecting the current state of the COTS-based MPoC Software.	
1C-2.3 The A&M data sent to the back-end implements methods to ensure the freshness and authenticity of the data.	**1C-2.3.a** The tester must confirm through examination and observation that: • Mechanisms exist that provide freshness. • The method used to ensure the A&M data (including the freshness indicator) is authentic—i.e., it is validated as coming from the expected source, and has not been tampered with during transmission.	The A&M needs to resist replay, preplay, or tampering attacks. The data providing freshness to the attestation system needs to be protected against tampering—e.g., the use of a signed timestamp or nonce known beforehand to the A&M back-end.

Security Requirements	Test Requirements	Guidance
1C-2.4 The A&M functionality includes an aspect within the COTS-based MPoC Software, which performs continual monitoring.	**1C-2.4.a** The tester must confirm through examination and observation that the A&M functions include checks that execute at all times when the COTS-based MPoC Software is executing to detect COTS operating modes that may impact security.	On-device attestation can be time and resource intensive, and therefore full attestation checks cannot be continually performed. However, some checks are less intensive—such as the check for entry into developer mode, or activation of debug functionality—and can be more easily performed in the background at all times. To ensure the security of the MPoC Solution, it is important that the checks which are more easily deployed at all times, are not left until a full attestation is required.

1C-3 Response

The A&M must be able to respond to potential attacks that it has identified. The response process, and the data that may lead to that response, must be documented. Attacks detected by the A&M of the COTS-based MPoC Software must be reported to the A&M back-end.

Security Requirements	Test Requirements	Guidance
Objective: The A&M must be able to provide suitable responses to mitigate potential attacks.		
1C-3.1 Documentation exists that describes the actions that can be taken if the attestation and monitoring (A&M) indicates that the COTS-based MPoC Software, or COTS platform is potentially compromised.	**1C-3.1.a** The tester must confirm through examination that the provided information is sufficient and consistent with the tester's understanding of the solution.	The A&M system needs to implement actions when tampering of the COTS-based MPoC Software is detected. Information outlining the functions provided by the attestation and monitoring (A&M) to indications of compromise is required. It is likely that the attestation and monitoring (A&M) back-end is able to be configured to act in different ways and, where this is possible, the documentation needs to reflect how such configuration is managed securely. Common responses from the attestation and monitoring (A&M) include account deactivation, assets deletion, and temporal suspension until a manual verification occurs. Checks that attempt to validate the integrity of the COTS platform are included in this requirement. Such checks may include validating that the COTS device has not been rooted, jailbroken, or otherwise compromised so that there is no longer a secure platform on which the COTS-based MPoC Software can execute.
	1C-3.1.b The tester must confirm through examination that the documentation includes potential configuration, or options for these actions must also be documented.	

Security Requirements	Test Requirements	Guidance
1C-3.2 Potential tampering events detected by the COTS-based MPoC Software are reported to the attestation and monitoring (A&M) back-end.	**1C-3.2.a** The tester must confirm through examination and observation that the COTS-based MPoC Software is able to report any potential tampering events to the A&M back-end.	If a tamper attempt is detected by the COTS-based MPoC Software, it is required that the COTS-based MPoC Software informs the A&M back-end. This can assist with the understanding of risk for similar types of solutions, as well as helping the A&M responses to be further tuned or developed to accommodate similar types of attacks. Additionally, as the COTS-based MPoC Software is considered to execute in a potentially hostile environment, communication of potential tamper events assists in aligning back-end responses with expected responses from the COTS-based MPoC Software. For example, if a COTS-based MPoC Software instance detects an attempted compromise and performs actions to halt payment processing, further payment processing from that instance received by the back-end system should be considered suspect.
1C-3.3 It is possible for the COTS-based MPoC Software to disable processing in the event of tamper indications.	**1C-3.3.a** The tester must confirm through examination and observation that the COTS-based MPoC Software is able to respond independently to potential tamper events detected by local A&M checks. **1C-3.3.b** The tester must attempt to trigger a subset of the security mechanisms of the COTS-based MPoC Software and verify that the actions taken by the MPoC Solution are as documented. This test must be performed both with and without connection to the A&M back-end to confirm that independent action is possible. ***Note:*** *This test is not to verify the attestation policy, but to verify that the documented response capabilities exist in the MPoC Solution and can be triggered.*	The ability to respond to potential attacks is a requirement even during any loss of communication with back-end A&Ms. Tampering events may include both direct attacks on the COTS-based MPoC Software, such as detection of overlay during PIN entry, as well as compromises of the integrity of the COTS platform, such as rooting, jailbreaking, etc.

Security Requirements	Test Requirements	Guidance
1C-3.4 The COTS-based MPoC Software performs an attestation with the A&M back-end upon start up and at least every 60 minutes of continuous operation or suspends further payment processing until such attestation is performed.	**1C-3.4.a** The tester must confirm through examination and observation that the COTS-based MPoC Software prevents further payment processing after 60 minutes of continuous operation without a passing response from the A&M back-end component when not operating in offline payment acceptance mode.	When not operating in offline payment-processing mode, the COTS-based MPoC Software is required to receive communications from the A&M back-end component at least every hour to ensure that the MPoC Solution is operating securely and that the COTS-based MPoC Software has passed all attestation checks successfully. In the context of this requirement, a "passing response" from the A&M back-end component indicates that the A&M system has performed and validated all attestation checks required by the MPoC Software attestation policy. This requirement does not supersede any other requirements for consistent A&M monitoring or checks to be performed on execution or prior to specific functions such as PIN entry. This requirement does not replace Requirement 1C-2.4, which notes the need for the COTS-based MPoC Software to perform continual and ongoing checks at all times (although these checks may be a subset of a "full" attestation process performed locally).
1C-3.5 The COTS-based MPoC Software that has been suspended or otherwise halted performs an A&M attestation prior to any payment processing.	**1C-3.5.a** The tester must confirm through examination and observation that the COTS-based MPoC Software resumed from a suspended, halted, or other mode that prevents execution, requires an A&M attestation process is performed prior to any payment processing.	An attacker may attempt to exploit periods of no execution, or suspension of the COTS-based MPoC Software, to perform attacks on the COTS platform. In such instances it is important the security and integrity of the platform is validated prior to any payment processing. When assessing this requirement, the tester should consider how the COTS platforms supported by the COTS platform baseline operate with regards to application switching and halting. Some platforms may allow for continued secure operation when an application is not currently in focus or may allow for multiple applications to have presence on the display at any one time—e.g., during multi-tasking or multi-window operation. *(continued on next page)*

Security Requirements	Test Requirements	Guidance
		The MPoC requirements allow for both remote and local attestation functions, and the type of attestation to be performed upon return to the COTS-based MPoC Software that has been removed from focus may depend upon the COTS platform features and attestation methods implemented in local and remote operation.
1C-3.6 A&M results that may require the cessation of payment acceptance are protected against manipulation during transmission to the payment back-end.	**1C-3.6.a** The tester must confirm through examination and observation that any A&M results that may require the cessation of payment acceptance are protected against manipulation during transmission to the payment back-end.	A&M results that indicate that the payment acceptance must be halted are likely to result from a compromised COTS device or COTS-based MPoC Software. This may impact the ability for the payment back-end to receive accurate and correct data from the COTS device. This requirement does not prevent or preclude operational models that provide the COTS-based MPoC Software with explicit approval to perform payments on a per-transaction basis. For the purposes of this requirement, blocking or otherwise preventing receipt of A&M results to the back-end—for the purposes of continuing payment processing on a compromised system—is considered manipulation.

1C-4 Anti-Tampering

The COTS-based MPoC Software is protected against tampering through the utility of the A&M system. For this reason, the attestation and monitoring (A&M) is a high-value target for attackers and becomes a critical target that needs to be protected against tampering.

Security Requirements	Test Requirements	Guidance
Objective: The attestation and monitoring (A&M), including the collection and transmission of data on the COTS device, is protected from compromise.		
1C-4.1 The protections provided to the attestation and monitoring (A&M) are documented.	**1C-4.1.a** The tester must confirm through examination that the information provided sufficiently details the anti-tamper protections that have been validated through the COTS-based MPoC Software testing process.	Modification of the attestation and monitoring (A&M) data or code may, in practice, allow for the bypass of the security mechanisms. The A&M component on the COTS-based MPoC Software needs to be protected against tampering. Documentation is required to indicate what protections are provided, so that these can be confirmed as in-place during the lab validation.
1C-4.2 The local time source used by the COTS-based MPoC Software is secured against tampering or alteration.	**1C-4.2.a** The tester must confirm through examination and observation that the local time source is protected against tampering.	The time source used to track the local time is required to be reliable and trusted. The COTS platforms and MPoC Solutions may offer different options of measuring time. The COTS-based MPoC Software is required to select the option that reflects real elapsed time to enforce the time limits properly. For example, time may be provided by a PCI PTS POI during each transaction. In these cases, the tester is expected to validate that interaction with the PCI PTS POI is required for each transaction. Timers used for this purpose need to be monotonic, and clock sources, if used, need to prevent or log alteration by applications or users. If the clock value could be affected by, or altered during, sleep events or periods when the COTS-based MPoC Software is not executing, payment processing must be prevented after such events until connectivity to the back-end system can be re-established and the offline payment data cleared. *(continued on next page)*

Security Requirements	Test Requirements	Guidance
	1C-4.2.b The tester must confirm through examination and observation that the local time source is validated during connections to the back-end A&Ms.	Validation of the local time source during connections to the attestation and monitoring (A&M) helps to identify attacks that may attempt to manipulate this source. Additionally, it helps to ensure that the attestation and monitoring (A&M) can correctly and accurately interpret any time stamps used in transmitted attestation and monitoring (A&M) data.
1C-4.3 The A&M back-end is able to detect failures in the A&M functions within the COTS-based MPoC Software.	**1C-4.3.a** The tester must confirm through examination and observation that the A&M back-end is able to detect and respond to failures in the COTS-based MPoC Software A&M functions.	The A&M back-end is required to monitor and be able to respond to situations where there is an indication of failure in the COTS-based MPoC Software. For example, if attestation data is repeated when it should not be, or is not present, when the COTS-based MPoC Software appears to be continuing to transact, it may be an indication of an attack on the payment system.
1C-4.4 The A&M used by the COTS-based MPoC Software is resistant to tampering to an attack rating of 25 points using the attack-costing framework in *Appendix B*.	**1C-4.4.a** The tester must test the COTS-based MPoC Software by attempting to tamper with the attestation system and the messages sent and received from the attestation and monitoring (A&M) back-end.	The A&M plays the important role of communicating the security state of the COTS and PCI PTS POI devices to the back-end and concentrating the security information of the COTS-based MPoC Software. A compromise of the attestation and monitoring (A&M) component on the COTS device or data transmitted to the back-end may be the same as effectively disabling security checks.
	1C-4.4.b If security mechanisms prevent the tampering, the tester must attempt to bypass the mechanisms. If it is not possible to bypass the mechanism, the tester must describe what would be needed to bypass the mechanism successfully (the expected actions/situations needed to bypass the mechanism).	
	1C-4.4.c The tester must provide a costing of this attack based on the method outlined in *Appendix B. Attack Costing Framework*. This requirement is passed if the most feasible attack cannot be costed for less than 25 points.	

1C-5 A&M Integration Guidance

The attestation and monitoring (A&M) needs to be integrated securely into the MPoC Solution. Therefore, it is important to ensure that information exists that facilitates the transfer of the security and operational knowledge required.

Security Requirements	Test Requirements	Guidance
Objective: Sufficient guidance and features are provided to allow for the secure integration and operation of the attestation and monitoring (A&M).		
1C-5.1 A&M back-end operation security guidance information that explains how the attestation and monitoring (A&M) is securely configured and operated exists.	**1C-5.1.a** The tester must confirm through examination that the attestation and monitoring (A&M) back-end operation security guidance contains the required information and comment on its sufficiency to securely configure the attestation and monitoring (A&M) back-end according to the tester's understanding of the MPoC Software. The documentation must include the following at a minimum: • How to configure the attestation and monitoring (A&M), including the offline policy, securely. • A base configuration for the attestation and monitoring (A&M). • Specific integration requirements that must be fulfilled to integrate the attestation and monitoring (A&M) component securely.	This information is required even when the attestation and monitoring (A&M) is operated by the developer of that system, as correct operation requires proper levels of training and documentation. It is not sufficient for operation of an attestation and monitoring (A&M) environment to rely entirely on experience and knowledge that is not otherwise documented.
	1C-5.1.b The tester must confirm through examination that the attestation and monitoring (A&M) security guidance includes a base configuration defined for the A&M component. This configuration is assumed to be a secure configuration for the A&M and must be the one used for the PCI MPoC validation tests.	

Security Requirements	Test Requirements	Guidance
1C-5.2 The security guidance information details how to securely integrate the A&M software component into the MPoC Application.	**1C-5.2.a** The tester must confirm through examination that the security guidance includes at a minimum: • How to deploy the A&M software component in a production-ready configuration—e.g., not debugging enabled. • How to integrate the A&M software component into the MPoC Application. • Specific dependencies of security checks on OS versions or COTS platforms.	Integration of the A&M functionality is required for all MPoC implementations. This security guidance may be a formal document for MPoC Software components that are to be operated by third parties, or internal knowledge base solutions for vendors performing all operations required of an MPoC Solution.
1C-5.3 Security guidance information details what A&M features are configurable, the processes for setting or changing these configurations, and how the applicable settings may affect the security and functionality of the overall MPoC Solution.	**1C-5.3.a** The tester must confirm through examination that the configuration options of the A&M are documented, including their possible settings. **1C-5.3.b** The tester must confirm through examination that the configuration options exist and are limited to the documented options.	An A&M can be configured at runtime or compile time. Documentation that clearly states what signals and responses are configurable, and which ones are not, help understand the baseline security of the solution and work as an input for policy making. Sufficient information needs to be provided over the configuration of the A&M component, such that it can be configured properly during integration and so the impact of such configuration decisions can be understood.

Module 1D: Secure Entry and Processing of Account Data

This Module covers the use of external and/or COTS-native systems to read payment account data from a payment instrument such as a payment card. This Module does not cover the entry of account data not obtained from the payment instrument, such as PINs. This Module includes requirements for the use of PCI PTS POI devices, non-PTS approved MSR devices, the use of the COTS-native NFC interface to read contactless payment instruments, and manual entry of account data.

Only account data-entry systems specifically addressed by this standard may be used within an MPoC Solution. Future updates to this standard may include or support other methods of account data entry.

1D-1 Account Data Entry and Encryption

This Section covers the requirements for all COTS-native account data-entry methods, and how this data must be secured as it is entered and processed. These requirements do not apply to data captured through a PCI PTS POI or non-PTS approved MSR device. PAN that is truncated in accordance with relevant PCI DSS FAQs is not considered in scope for these requirements.

Security Requirements	Test Requirements	Guidance
Objective: All account data-entry methods are documented and provide encryption for onward processing.		
1D-1.1 Documentation exists that details how account data is entered and secured.	**1D-1.1.a** For each of the account data-entry methods supported by the solution, the tester must confirm through examination that the required information for integration with the COTS-based MPoC Software is present. The information must provide details about: • All methods used for the entry of account data into the COTS-based MPoC Software. • The supported functionality of each account data-entry method. • All guidance required for the correct and approved operation of any of the account data-entry methods.	It is a requirement that all account data-entry methods that the COTS-based MPoC Software supports are understood and secured properly. Use of functionality that has not been assessed presents a security risk to the COTS-based MPoC Software. Documentation associated with any external or add-on account data-entry methods needs to contain all the necessary information to integrate that system securely into the MPoC Solution.

Security Requirements	Test Requirements	Guidance
1D-1.2 Account data is encrypted at the earliest possible point.	**1D-1.2.a** The tester must confirm through examination that for any account data-entry methods implemented on the COTS device, the card data is encrypted either immediately upon entry or immediately after any required processing through associated payment kernels.	Account data is required to be encrypted at the earliest to reduce the opportunity window in which account data can be compromised in cleartext. Normally, this is done in the external card reader itself or in the COTS-based MPoC Software when a COTS native interface, such as COTS-native NFC, is used. Although it is not the scope of this requirement to assess the encryption of any external account data capture systems, such as a PCI PTS POI or non-PTS approved MSR, the assessor is required to confirm that there is no functionality on the COTS device that would expect this data to be provided in cleartext or allow for the decryption of the account data on the COTS platform. Account data encryption is required to be provided separately from any secure channels provided. Encryption of account data by the secure channel, such as through the use of TLS, is not sufficient to meet this requirement. Datagram-level encryption using encryption keys dedicated to account data encryption needs to be used.
1D-1.3 Account data provided from acceptance devices external to the COTS device (such as from a PCI PTS POI or non-PTS approved MSR device) must be provided encrypted and not be decrypted on the COTS platform.	**1D-1.3.a** The tester must confirm through examination and observation that account data provided by acceptance devices external to the COTS device, such as PCI PTS POIs or non-PTS approved MSRs, is received encrypted and the cryptographic keys required to decrypt this data are not available on the COTS device, and the cleartext account data is never returned to the COTS device.	PTS POI devices that are listed as validated to the SCRP approval class will always ensure that account data is encrypted prior to transmission to another system, such as to the COTS-based MPoC Software. Other PCI PTS POI devices may not always encrypt such data, even if those devices are listed as providing SRED functions. One method of validating that such devices will provide encryption of account data at all times is to ensure that those devices and the applications they execute are part of a listed PCI P2PE solution.

Security Requirements	Test Requirements	Guidance
1D-1.4 The COTS-based MPoC Software truncates or masks the PAN, using methods compliant to relevant PCI DSS FAQs, when outputting or displaying PAN data that is not encrypted.	**1D-1.4.a** The tester must confirm through examination and observation that the PAN is appropriately truncated or masked when output or displayed—e.g., by using the COTS device screen or printing a receipt—according to the relevant PCI DSS FAQs.	Customer receipts may be necessary for customer validation of the transaction and as part of a formal payment challenge process. When such data is sent from the back-end environment for display or printing in the merchant environment, any PAN data on the receipt is required to be truncated to ensure that it cannot be uniquely correlated with the customer and that full details are not available to the merchant. Functions to display cardholder data, either during or post transaction processing, could be used to facilitate the disclosure and compromise of this data. The COTS-based MPoC Software can display PAN data that is truncated or masked as per relevant PCI DSS FAQs. The intent of truncation is to remove a segment of PAN data permanently, so that only a portion of the PAN is available on the COTS device.
	1D-1.4.b The tester must confirm through examination and observation that the MPoC Software does not provide functions to display cleartext cardholder data that is not truncated or masked.	

Security Requirements	Test Requirements	Guidance
1D-1.5 The COTS-based MPoC Software is able to detect and respond to events that may impact the security of the account data-entry process.	**1D-1.5.a** The tester must confirm through examination and observation that the following events which may impact the security of the account data-entry process are able to be detected: • The COTS-based MPoC Software pauses or stops executing. • The COTS-based MPoC Software loses its foreground focus. • An external card reader provides cleartext account data.	The transaction process needs to be protected against manipulation or subversion. Attempts to modify or overlay the cardholder prompts—e.g., instructions to the cardholder—or other UI features that are important for the security of the solution are required to be prevented. Pausing the application usually means that the application remains partially visible while the user interacts with a different dialog or screen—e.g., in multi-screen mode. When the user switches to a different application, the application is considered "stopped" until either the user switches back or the system destroys the instance of the application. Although account data entered through external devices like a PCI PTS SCRP is encrypted prior to transmission outside of that device, designs need to consider the security implications of pausing or changing application focus. For example, interception or interruption of the communications path between the MPoC Software and a PCI PTS SCRP should not allow for compromise of any account data or access to sensitive functions within the external card acceptance device. Additionally, a PCI PTS POI device that is not an SCRP may output account data that is not encrypted, and the SDK should be able to detect and respond to such occurrences. The security of the other applications on the COTS device is not known and therefore it needs to be assumed that the COTS environment may be hostile. In the context of this requirement, a "response" is implemented based on the attestation policy of the MPoC Solution. It is expected that where an external reader provides cleartext account data the event should be escalated and result in the disabling of that reader until it can be assured that this type of event will no longer occur.

Security Requirements	Test Requirements	Guidance
1D-1.6 The COTS-based MPoC Software does not store account data beyond the completion of the current transaction process for any purposes other than offline payment processing.	**1D-1.6.a** The tester must confirm through examination and observation that account data is not stored on persistent storage, beyond the current transaction process, except for the purposes of offline payment processing.	Account data is permitted to be stored on the COTS device only for the purposes of offline payments. This includes storage where the account data may be protected through means such as encryption or storage in tamper-resistant processing elements. For the purposes of this requirement, “storage” refers to any non-volatile or long-term memory, such as Flash or RAM discs. Temporary storage of data within the memory of the COTS devices is permitted; however, any such storage needs to be cleared securely as soon as possible—no later than when the transaction has been sent for processing. Support for offline processing and associated data storage is not mandatory and may be conditional or disabled as required. In the context of this requirement offline payment processing is defined as per Section 1F.

1D-2 Use of PCI PTS POI-approved Devices

If an external card reader is used to accept chip-based transactions, or any type of contact-chip based transactions are to be accepted, the card reader used must be validated and listed as a PCI PTS POI. A PCI PTS POI must be used for any MPoC Solution that accepts contact chip-based payment cards, as well as for any offline PIN-based transactions.

Security Requirements	Test Requirements	Guidance
Objective: Secure card readers supported by the solution provide sufficient protection to account data.		
1D-2.1 The security guidance document details the secure card readers supported by the MPoC Software.	**1D-2.1.a** For each of the PCI PTS devices used in the solution, the tester must confirm through examination that, at a minimum, the following information is provided within the security guidance document assessed under Section 1G: • Hardware, firmware, and listing details for each PCI PTS device used. • Supported functionality of each secure card reader. • All guidance required for the correct and approved operation of the used secure card reader.	Any PCI PTS devices used with the COTS-based MPoC Software need to have been assessed for the specific use implemented in the COTS-based MPoC Software. Use of functionality that has not been assessed presents a security risk to the COTS-based MPoC Software. The documentation provided by the MPoC Software vendor needs to contain all the necessary information to integrate the PCI PTS POI securely into an MPoC Solution.
1D-2.2 Any chip accepting devices are approved to the PCI PTS POI requirements.	**1D-2.2.a** The tester must document all chip accepting devices that can be used with the COTS-based MPoC Software, including the PCI PTS listing number and confirm that each is approved for the chip-acceptance methods used by the MPoC Software.	Chip accepting devices include any devices accepting contact or contactless chip cards. Approval to the PCI PTS POI requirements may be validated through listing on the PCI SSC website. Other requirements may apply to any PCI PTS POI devices used, depending on the implementation.

Security Requirements	Test Requirements	Guidance
1D-2.3 The PCI PTS POI devices are attested as part of the COTS-based MPoC Software attestation system including validation that the device is operating in an encrypting mode that prevents transmission of cleartext account data to the COTS device.	**1D-2.3.a** The tester must confirm through examination and observation of the A&M that: • Any PCI PTS devices used are uniquely identified and validated. • The firmware version is verified by the A&M. • The state of the PCI PTS device is verified by the A&M. • PCI PTS devices have an approved version of firmware installed, as listed on the PCI PTS listing for that device. • The device is operating in an encrypting mode that prevents cleartext account data being transmitted to the COTS device.	The A&M needs to ensure that all aspects of the COTS-based MPoC Software are operating as expected and are not in a state that indicates or could facilitate compromise. This includes validating the state and security posture of any attached card-reading devices. This validation included confirmation of the firmware version(s) of the attached devices, as well as hardware versions, and confirming that these meet the expectations of the implementation—e.g., they are up to date with any relevant patches. PCI PTS POI devices listed as validated to the SCRP approval class provide for the use of "enablement tokens" which can be used to tie the continued operation of these devices to positive A&M results. Use of enablement tokens with an MPoC implementation is recommended as best practice. Refer to the MPoC Program Guide for details on acceptable PCI PTS POI versions.
1D-2.4 Whitelists allowing for the exposure of cleartext data from a PCI PTS POI device are: • Cryptographic authentication by the PCI PTS POI device's firmware. • Only allow for the output of non-PCI payment brand card data. • Documented and justified.	**1D-2.4.a** The tester must confirm through examination and observation that any whitelisting that allows for the exposure of cleartext card data from a PCI PTS POI device must be: • Cryptographic authentication by the PCI PTS POI device's firmware. • Only allow for the output of non-PCI payment brand card data. • Documented and justified.	Whitelisting may be supported by some PCI PTS POI devices, allowing for the output of cleartext data from cards presented to those devices. Such whitelists may be used to capture and read non-payment cards such as merchant ID cards or loyalty cards. However, this requirement does not overrule requirement 1D-1.3 which requires that all account data submitted from an external reader (including a POI device) is provided encrypted (for PCI brand cards).

1D-3 Magnetic-Stripe Data

Magnetic-stripe data may be read only through external devices listed as meeting the PCI PTS POI requirements or as a Non-PTS approved MSR validated to the requirements of Appendix E.

Security Requirements	Test Requirements	Guidance
Objective: Data from magnetic-stripe cards is secured through approved readers.		
1D-3.1 The security guidance document details the MSRs supported by the MPoC Software.	**1D-3.1.a** For each of the devices used to accept magnetic-stripe transactions in the solution, the tester must confirm through examination that, at a minimum, the following details are provided in the security guidance document assessed under Section 1G: • The hardware, firmware, and validation details for each MSR used. • The supported functionality of each MSR. • All guidance required for the correct and approved operation of the used MSRs.	Any MSR devices used with the COTS-based MPoC Software need to have been assessed for the specific use implemented in the COTS-based MPoC Software. Use of functionality that has not been assessed presents a security risk to the COTS-based MPoC Software. The documentation provided by the MPoC Software needs to contain all the necessary information to integrate any devices accepting magnetic-stripe-based transactions securely into an MPoC Solution. This may include the use of PCI PTS POI devices as well as non-PTS approved MSR devices. Validation details may include reference to listing on a PCI validation list or assessment through the MSR Appendix.
1D-3.2 Magnetic-stripe cards are accepted only through readers listed as PCI PTS POI devices or are validated as a Non-PTS Approved MSR.	**1D-3.2.a** When magnetic-stripe transactions are supported, the tester must confirm through examination and observation that only approved PCI PTS POI or non-PTS approved MSR readers are able to be used to accept magnetic-stripe transactions that are processed online.	The COTS-based MPoC Software may accept magnetic-stripe-based transactions using either a PCI PTS POI that integrates an MSR, or a dedicated Non-PTS approved MSR device.
	1D-3.2.b The tester must document the validation and listing for any MSR Reader devices used with the COTS-based MPoC Software. Requirements for non-PTS approved MSR devices are provided in *Appendix E: MSR Security Requirements*.	

Security Requirements	Test Requirements	Guidance
1D-3.3 The MSR devices must be attested as part of the COTS-based MPoC Software attestation system.	**1D-3.3.a** The tester must confirm through examination that: • Any MSR device(s) used are uniquely identified and validated by the attestation and monitoring (A&M), including the firmware and hardware versions where applicable. • The state of the MSR device is verified by the attestation and monitoring (A&M). • The MSR devices(s) are confirmed by the attestation and monitoring (A&M) to have no known or exploitable vulnerabilities.	The attestation and monitoring (A&M) needs to ensure that all aspects of the COTS-based MPoC Software are operating as expected and are not in a state that indicates or could facilitate compromise. This includes validating the state and security posture of any attached card-reading devices. Firmware in the context of this requirement may include operational code that is executed by an underlying processor, or the version of a dedicated ASIC used in place of a general-purpose processor.
1D-3.4 MSR data captured in an MPoC Solution is not made available in cleartext on the COTS device.	**1D-3.4.a** The tester must confirm through observation and examination that any MSR data captured is never exposed in cleartext within the COTS device. This includes validating that no optional configuration settings or functions exist that may disable or prevent encryption from the MSR itself.	Data from a magnetic-stripe card can be easily copied and replicated onto a “cloned” card for use in fraudulent transactions. To help mitigate against this risk, the data from the magnetic-stripe is encrypted within the PCI PTS POI or non-PTS approved MSR device used and is not exposed in the COTS device.

1D-4 COTS-Native NFC Interface

Contactless payment instruments may be accepted through the COTS-native NFC interface of the COTS device. This interface may be open to all applications on the COTS device. Protections must be implemented to secure the account data as it is entered and to ensure that the data is encrypted as soon as possible after processing within any payment kernel.

The contactless kernel has direct access to assets. Therefore, the processing, memory, and storage used by this kernel must be protected to prevent the assets from being compromised when handled by the payment kernel.

The requirements in this Section apply to the COTS-native NFC interface only.

Security Requirements	Test Requirements	Guidance
Objective: Account data is protected and securely processed when it is read by the COTS-native NFC interface.		
1D-4.1 Information that details the implementation of the COTS-native NFC acceptance method, including the implementation for any contactless kernels, exists.	**1D-4.1.a** The tester must confirm the following through examination: • That all COTS-native NFC acceptance methods are detailed. • If part of the kernel is executed remotely (e.g., a cloud-based kernel), where and what parts of the kernel are included in this remote execution. • How the kernel is configured and how the configuration is protected. • How the kernel is secured against tampering.	Contactless kernels used by the COTS-based MPoC Software provide for processing of account data. Therefore, the exact implementation and integration are important to the overall security of the MPoC Solution. Contactless kernels need to be integrated in a way that ensures they are protected against tampering—e.g., by being integrated as part of the overall tamper protection and response of the COTS-based MPoC Software.

Security Requirements	Test Requirements	Guidance
1D-4.2 The COTS-based MPoC Software ensures that the COTS-native NFC interface is not accessed by other applications during a payment transaction.	**1D-4.2.a** The tester must confirm through examination and observation that the COTS-based MPoC Software implements mechanisms to prevent, monitor, or otherwise inhibit access to the COTS-native NFC interface by other applications during a payment transaction.	To prevent interference or interception of the COTS-native NFC communication, the COTS-based MPoC Software needs to attempt to implement methods to prevent or detect access to this interface by other applications during the payment transaction process. This may involve attempting to assert exclusive control to the COTS-native NFC interface before initiating the transaction or monitoring if another application accesses the interface. Alternatively, exclusive operation may be ensured to any foreground application by the COTS platform. The COTS-based MPoC Software needs to prevent transaction processing if it cannot provide a sufficient level of security to the COTS-native NFC interface.
1D-4.3 The COTS-based MPoC Software ensures that the COTS device camera(s) are not accessed by other applications during a payment transaction where presentment of the card may be captured on the COTS camera.	**1D-4.3.a** The tester must confirm through examination and observation that the COTS-based MPoC Software implements mechanisms to prevent, monitor, or otherwise inhibit access to the COTS device camera(s) by other applications during a payment transaction where presentment of the card may be captured on the COTS camera.	The camera can be used as a side-channel source to gain access to assets such as CVC/CVV. To prevent unauthorized visual capturing of account data from a payment instrument, such as a payment card when it is in proximity to the COTS device, the COTS-based MPoC Software should attempt to prevent other applications and processes running on the COTS device from using the camera. The expectation is that other applications on the COTS device are not able to use the camera when the COTS-based MPoC Software prompts the cardholder to initiate a contactless payment, or that access by another application during this time is detected and the payment transaction halted. Transactions which use external devices (such as a PCI PTS POI) to capture the cardholder data, such that it is infeasible to use the COTS camera to capture an image of the card, do not need to meet this requirement.

Security Requirements	Test Requirements	Guidance
1D-4.4 When part of the kernel functionality is implemented remotely the connection between the COTS-based MPoC Software and the remote component must be protected using a secure channel, and relevant requirements of Domain 4 and Domain 5 are met.	**1D-4.4.a** The tester must confirm through examination that the communication with remote components of the kernel is secure through use of secure channels, and relevant requirements of Domain 4 and Domain 5 are met.	For account data, protection using a secure channel is not considered sufficient, application-level encryption is required. Remote kernel implementations are expected to manage account data, at least including PAN, and therefore are required to be validated to PCI DSS.

1D-5 Manual Entry

Transactions using account data obtained from a manual entry process may be processed by an MPoC Solution only when no PIN entry is performed, and when the manual data-entry process is protected. Entry of truncated PAN data, or data into software not executed on the COTS device, is not included in scope of these requirements.

PAN that is truncated in accordance with relevant PCI DSS FAQs is not considered in scope for these requirements.

Security Requirements	Test Requirements	Guidance
Objective: Account data entered into the COTS device through manual entry methods is secured.		
1D-5.1 Documentation exists that details the manual entry process and protection methods.	**1D-5.1.a** For each of the manual account data-entry methods used in the solution, the tester must confirm through examination that details about the following are provided at a minimum: • The ways in which the manual entry method can be invoked. • The protections applied to the manual entry method. • The types of account data that may be manually entered.	The COTS-based MPoC Software may support manual entry of the PAN, e.g., to enable payment processing if other card presentment modes fail (due to a damaged chip or magnetic-stripe). Additional account data, such as the security code for the card, may be additionally required during such payment processing. It is important that any such methods of manual entry are detailed to allow for a complete understanding and security assessment of these modes of data presentment.
1D-5.2 Manually entered account data is used only for the purposes of transaction processing.	**1D-5.2.a** The tester must confirm through examination and observation that manually entered account data is used only for the purposes of transaction processing.	The COTS-based MPoC Software is permitted to provide manual entry for account data only for the purposes of transaction processing. For example, manual entry of PAN data as a data element in a search of prior transactions is not permitted.

Security Requirements	Test Requirements	Guidance
1D-5.3 Manually entered account data is protected during entry.	**1D-5.3.a** The tester must confirm through examination and observation the following for each of the manual account data-entry methods implemented: • The entry methods do not allow the storage of the account data. This includes by the OS for purposes such as spell-checking dictionary creation. • The account data is entered directly into the COTS-based MPoC Software. Third-party keyboard or data-entry applications are not used for account data entry. • It is not possible to take screenshots or recordings of the COTS-based MPoC Software during PAN entry on the COTS device into which the PAN is entered.	It is common in many COTS OS for the OS-provided data-entry functions to store data entered so that it can build a dictionary of commonly used terms for the purposes of enhancing spell-check functions. In addition, some OSs allow for the replacement of the OS provided data-entry functions with third-party applications that may provide similar functions or perform other operations with data that is entered. It is important that account data-entry functions secure against the exposure of the sensitive assets entered through these types of systems. To prevent oversight of the account data during entry, no more than one digit may be displayed at any time. Common implementations may display the more recently entered digit, with all other digits obfuscated using a generic symbol such as an asterisk. Capture of the account data through optical means, such as by taking a photograph of the payment card, is also to be secured in line with this requirement, where implemented.
1D-5.4 The COTS-based MPoC Software is able to detect events that could impact the security of the entry process.	**1D-5.4.a** The tester must confirm through examination and observation that mechanisms exist to detect events that could impact the security of the entry process, and that there is a defined process for how to handle such events.	Common malware attacks are known to overlay information on top of legitimate applications to obtain input data from the user. However, an overlay may not always be a sign of an attack, so a defined process on how to handle such events is required.

Module 1E: PIN Entry on COTS Device

This Module covers COTS-native PIN entry, even if the capability can be disabled at runtime with configuration options. If the COTS-based MPoC Software does not support PIN entry (or any other situation where the PIN may be exposed) on the COTS devices, this Module is not in scope. It is possible that the numeric value of the PIN is not captured by the COTS-based MPoC Software directly; instead, some interim value may be collected, such as touch locations or different numeric values, which must be reconstructed into the PIN at a later time by the COTS-based MPoC Software. These interim values must also meet all requirements of this Section in respect to protections prior to the construction of the encrypted PIN block. In all cases, values related to the customer PIN, including individual digits, touch locations, or numeric values that can be mapped back to PIN values, must never leave the COTS-based MPoC Software until formed into a fully encrypted ISO format 4 PIN block.

An MPoC Software Product or MPoC Solution can support PIN entry only through use of the touch screen on the COTS platforms supported, or through use of a PIN entry device listed as validated to the PCI PTS POI standard. Use of physical keypads for PIN entry on non-tamper responsive devices is not permitted.

Accessibility features may be made available on a per-transaction basis and must not be the default or sole PIN entry method offered. Accessibility features must not display the individual PIN digits themselves or provide feedback (audio, visual, or haptic) that is unique to individual PIN digits. “Zoom” features to increase the size of keypad buttons may be provided, as long as individual PIN digits cannot be uniquely identified.

Side channel testing (Requirement 1E-1.3) of the accessible PIN entry mode is not required. The MPoC Software and the A&M system must be designed to protect any unique features of the accessible PIN entry process, including monitoring and tracking the number of times accessible PIN entry is used. When enabled, accessibility features must provide clear indication that they have been enabled and allow for the cardholder entering the PIN to disable this mode at any time. Accessibility features must exit upon completion of the payment transaction process, including error states, and require explicit enabling for each separate payment transaction.

1E-1 COTS-native PIN Entry

This Section covers the entry of cardholder PINs on the COTS device. When the PIN is entered on the COTS device, regardless of whether the PIN entry mechanisms are backed by hardware (e.g., TEE), the COTS-based MPoC Software is required to comply with this Section.

Security Requirements	Test Requirements	Guidance
Objective: Cardholder PINs are protected as they are entered and processed on the COTS device.		
1E-1.1 Documentation exists that describes the secure capture and processing of the cardholder PIN.	**1E-1.1.a** The tester must confirm through examination that the information provided contains, but is not limited to: • How the PIN is protected during entry and prior to encryption. • How the PIN is encrypted. • The security mechanisms implemented to prevent the extraction of the PIN at runtime. • Side-channel prevention mechanisms. • The types of PIN-verification method(s) supported (offline, online, or both). • Any accessible PIN entry modes that are implemented.	COTS-based MPoC Software that accepts account data entry, including PINs, on COTS devices has to provide a path for these sensitive assets from the hardware of the COTS device, through the OS and drivers, to the COTS-based MPoC Software itself. Documentation is required to demonstrate an understanding and consideration for the entire path taken by this data, not just how the data is received and processed within the COTS-based MPoC Software. For example, protections need to consider how to prevent determination of PINs through OS-level features such as screen recording or logging of touch inputs. The PIN needs to be protected against extraction using side-channels analysis (e.g., using the accelerometers and gyroscopes to obtain the coordinates of user touch events and correlate such touch events to individual PIN digits).
1E-1.2 PIN entry is supported only for chip-based transactions.	**1E-1.2.a** The tester must confirm through examination and observation that PIN entry is supported only for transactions with chip-based cards.	Magnetic-stripe cards do not provide the same security features present in chip-based cards and can be easily "cloned" or copied. COTS-based MPoC Software that provides for the capture of both PIN and magnetic-stripe track data for the same transaction are not permitted. This is true even if the track data is sent to the COTS device encrypted, from an attached POI or Non-PTS approved MSR, because the data may be captured through a skimmer on the peripheral reader device itself.

Security Requirements	Test Requirements	Guidance
1E-1.3 The COTS-based MPoC Software does not leak complete or partial PIN digits. The COTS-based MPoC Software protects against side channels that use sensors present in the COTS device—e.g., accelerometers and gyroscopes—and screen capture.	**1E-1.3.a** The tester must confirm through examination that: • The threat of extracting the PIN using the COTS device sensors was considered. • Each sensor has a rationale on why they do not present a risk for side channel extraction of the PIN. • Measures were implemented to prevent the leakage of the PIN using the sensors that present a risk.	Some COTS device sensors can be used at the time of PIN entry to extract the PIN that is being entered by the cardholder. Furthermore, a COTS device output—e.g., distinctive sounds per key or change in the key digit on screen when pressed—can be used to extract the PIN from the COTS device using screenshots or recordings. Therefore, the COTS-based MPoC Software needs to ensure that the implementation accommodates for these issues and ensures that the method of PIN entry does not leak PIN digits through potential side channels. The use of a scrambled keypad may assist with meeting this requirement, but use of such a scrambled keypad may not be necessary if other mitigations are applied (such as disabling sensors during PIN entry, or use of a PIN entry keypad that appears in different locations for each PIN entry process). Accessible PIN entry modes are not required to be validated through testing to this requirement.
	1E-1.3.b The tester must confirm through examination, observation, and testing the correct implementation of the measures that mitigate the extraction of the PIN through side channels.	
1E-1.4 The COTS-based MPoC Software protects the PIN digits during entry.	**1E-1.4.a** The tester must confirm through examination and observation that: • There is no feedback that could be used to identify the PIN digits. • It is not possible to take screenshots or recordings of the COTS-based MPoC Software during PIN entry on the COTS device into which the PIN is entered. • Where a physical keypad is used for PIN entry, that keypad is part of a device listed as validated to the PCI PTS POI requirements.	Some COTS device sensors can be used at the time of PIN entry to extract the PIN that is being entered by the cardholder. Furthermore, a COTS device output—e.g., change in the key digit on screen when pressed—can be used to extract the PIN from the COTS device using screenshots or recordings. Therefore, the COTS-based MPoC Software needs to ensure that the implementation accommodates for these issues and ensures that the method of PIN entry does not leak PIN digits through potential side channels.

Security Requirements	Test Requirements	Guidance
1E-1.5 The PIN is encrypted into an ISO format 4-PIN block as soon as it is captured, and prior to export from the COTS-based MPoC Software and COTS device.	**1E-1.5.a** The tester must confirm through examination and observation that: • The PIN is encrypted into an ISO format 4-PIN block upon entry, and in all cases prior to transmission outside of the boundary of the COTS-based MPoC Software and COTS device. • If the cryptographic keys used to encrypt the PIN block are exposed in the rich execution environment of the COTS device, the PIN block is encrypted using unique keys per transaction.	Methods that capture and encrypt the PIN in ways not compliant to ISO9564 can result in unforeseen risk being introduced into the PIN-processing system. ISO 9564 Format 4 uses AES and encapsulates the PIN and PAN into the PIN block using separate encryption processes. This separation of the encryption processes within an ISO format 4-PIN block allows for the PIN and PAN to be managed as separate items during the formatting of the PIN block (on the COTS device). Maintaining separation between the PIN and PAN can help increase the overall security of the MPoC Solution. However, separation of the PIN and PAN is not a requirement. Only complete and encrypted ISO format 4 PIN blocks may be transmitted from the COTS device. A tokenized PAN, cryptographically bound to the actual funding PAN, may be used with these formats to allow for more complete dislocation of the PIN and PAN within the COTS device. Use of a PAN token is not required. Use of a PIN-encryption key that is unique per transaction ensures that any compromise of a specific PIN entry event cannot be used to compromise previous PINs processed by that system. Only the entity that is intended to decrypt the PIN is permitted to have access to the decryption key. If a PCI PTS POI is used for offline PIN transactions, only the PCI PTS POI is permitted to decrypt a PIN after encryption by the COTS-based MPoC Software, prior to transmission to the payment card. If no PCI PTS POI is used, only the processing back-end is permitted to decrypt the PIN. PINs may be translated from ISO format 4 by a PCI PTS SCRP or by back-end systems, for onward processing as required. PCI PTS devices not validated to the SCRP approval class have not been confirmed to provide secure PIN translation and may not be used for this purpose.

Security Requirements	Test Requirements	Guidance
1E-1.6 Attestation functions detecting indications of potential compromise are executed prior to each PIN entry process.	**1E-1.6.a** The tester must confirm through examination and observation that attestation is performed prior to each PIN entry.	PIN entry is permitted only when the COTS-based MPoC Software is running in a secure state. It is important that these attestation functions be performed before the cardholder is prompted to enter PIN data. The COTS-based MPoC Software may have multiple levels of attestation and is required to have some level that is always active. Due to power and processing constraints, it may not be possible to have all attestation functions always active, and so execution of more complete attestation prior to PIN entry is required. Validation of attestation data is expected to be performed online for any online PIN entry process.
	1E-1.6.b The tester must confirm through examination and observation that attestation and monitoring (A&M) validation is performed on the COTS device immediately prior to PIN entry.	
1E-1.7 The COTS-based MPoC Software detects when another application overlays, shares the screen during PIN capture, or otherwise could impact the security of the PIN entry process. In case of positive detection of events that could impact the security of PIN entry, the COTS-based MPoC Software cancels any transaction currently in progress and provides notification of this event to the back-end A&M system.	**1E-1.7.a** The tester must confirm through examination and observation that mechanisms exist to detect when another application overlays the screen, shares the screen, or otherwise could impact the security of the PIN entry process. In the case where an event is detected that could impact the security of PIN entry, the transaction currently in progress must be cancelled and a notification of this event must be provided to the back-end A&M system.	Common malware attacks are known to overlay information on top of legitimate applications to obtain input data from the user. The COTS-based MPoC Software needs to be able to detect whether another application is overlaying the COTS-based MPoC Software and stop any transaction in progress. As halting the current transaction may not be sufficient to prevent any in-progress attack, any transaction cancellation events are to be communicated to the back-end A&M so they may be used in future decisions about transaction processing.

Security Requirements	Test Requirements	Guidance
1E-1.8 PIN-related data (PIN, PIN related values such as touch locations, PIN block, PIN key) are not stored in the persistent storage and are erased once no longer required.	**1E-1.8.a** The tester must confirm through examination and observation that PIN-related data is not stored in persistent storage.	Cleartext PINs, encrypted PIN blocks, and PIN-encryption keys need to be cleared from the system as soon as they are no longer required. For both cleartext PINs and the PIN encryption key, this means once the encrypted PIN block is produced, these values are required to be securely erased. The COTS-based MPoC Software may perform the two encryptions required for an ISO format 4-PIN block as separate parts of the transaction, enabling encryption of the PIN as soon as possible and prior to the inclusion of the PAN or PAN token into the PIN block. The encrypted PIN block needs to be erased once the PIN block has been transmitted from the COTS device.
1E-1.9 Offline PIN verification is supported only through the use of PCI PTS POI devices which are approved for this purpose.	**1E-1.9.a** The tester must confirm through examination and observation that offline PIN verification is only allowed for transactions where the card has been accepted through a PCI PTS POI device that is listed as supporting offline PIN entry.	Offline PINs may be sent to the payment card in the clear, and therefore may be transmitted only when the card is presented into a tamper-responsive device, such as a PCI PTS POI.

Security Requirements	Test Requirements	Guidance
1E-1.10 Accessible PIN entry modes are implemented securely.	**1E-1.10.a** The tester must confirm through examination and observation that any accessible PIN entry modes provided are implemented securely. This includes: • Accessible PIN entry modes are not the default or sole mode of PIN entry. • Individual PIN entry digits are not highlighted or exposed through features such as screen zooming. • The accessible PIN entry mode provides a clear visual indication when it is enabled, and allows for the cardholder to disable this mode at any time. • The A&M functions are designed to protect the unique features of the accessible PIN entry process, including monitoring and tracking of the number of times the accessible PIN entry method is used. • Accessibility features exit on completion of the payment transaction process, including any error states, and require explicit enabling for each separate payment transaction.	Accessible PIN entry features are not required but may be made available on a per-transaction basis. They are not to be the default or sole PIN entry method offered. Accessibility features must not display the individual PIN digits themselves or provide feedback (audio, visual, or haptic) that is unique to individual PIN digits. “Zoom” features to increase the size of keypad buttons may be provided, as long as individual PIN digits cannot be uniquely identified.

Security Requirements	Test Requirements	Guidance
1E-1.11 Where PIN entry is performed on a device other than the COTS device: • That device is listed as validated to the PCI PTS POI requirements. • An approved PIN entry function of the device is used, ensuring the PIN is encrypted into an ISO 9564 compliant PIN block before leaving the PCI PTS POI device. • The PIN is not exposed in cleartext on the COTS device.	**1E-1.11.a** The tester must confirm through examination and observation that where a PIN may be entered through a device other than the COTS device: • That device is listed as validated to the PCI PTS POI requirements. • An approved PIN entry function of the device is used, ensuring the PIN is encrypted into an ISO 9564 compliant PIN block before leaving the PCI PTS POI device. • The PIN is not exposed in cleartext on the COTS device.	PCI PTS POI devices have been validated to provide secure entry of PIN data, however the applications used on such devices may influence how they operate. Many POI devices do not support encryption of PINs where the PAN is not directly entered into the POI device. If PIN entry is to be supported through an external device, that device is required to be listed as validated to the PCI PTS POI requirements, but also the laboratory is required to validate that an approved PIN entry function is being used for PIN encryption to ensure that this process has been correctly validated. If the PAN is to be transmitted to the POI for construction of the PIN block, the PAN must be secured in line with requirements of Section 1D of this standard. The COTS-based MPoC Software functions used to secure and transmit the PAN, including key management, is to be included in scope of assessment under Section 1A.
1E-1.12 PAN tokens, where used, are cryptographically bound to the PAN.	**1E-1.12.a** The tester must confirm through examination and observation that PAN tokens, where used, are cryptographically bound to the PAN.	Use of PAN tokens is not required but may be implemented as necessary—for example, to support COTS-native PIN entry with card acceptance on an external device which does not provide cleartext cardholder data to the COTS device. ISO 9564 format 4 PIN blocks allow for the use of PAN tokens in place of PANs, but these tokens must be cryptographically bound to the PAN to mitigate PAN replacement and PIN dictionary attacks.

Module 1F: Offline Payment Transactions

This Module is applicable to MPoC Software that supports offline transactions, even if the capability can be disabled at runtime with configuration options. Offline transactions covered by this Module are offline EMV transactions (where the transaction authorization is completely offline) and store-and-forward transactions (where the terminal approves the transaction, but the authorization is performed online after connectivity is regained).

Depending on the type of transaction, different types of data may need to be stored on the COTS device for eventual transaction authorization. Types of data that may need to be stored and protected during an offline transaction includes account data and transaction results (cryptograms).

Offline PIN verification (with a contact chip card presented through a PCI PTS POI) may be used in either online or offline approval processes.

Assets stored for offline transactions must be protected according to their security needs. This Module covers additional security requirements that are needed to provide protection to these assets when stored on the COTS device. The protection of assets during the processing of a payment is covered by separate Modules.

Support for offline payment processing is not a required feature of an MPoC Solution and may not be possible in some payment solutions due to local or payment brand specific rules.

1F-1 Offline Payment Transactions

This Section covers the requirements for securing the processing and storage of data required as part of offline payment transactions.

Security Requirements	Test Requirements	Guidance
Objective: Offline transaction results and account data are protected.		
1F-1.1 Stored assets, such as account data and transaction results, needed to process offline Payment transactions are encrypted in such a way that the cleartext values cannot be recovered on the COTS device after encryption.	**1F-1.1.a** The tester must confirm through examination, observation, and testing that any assets, such as account data and transaction results, stored for offline transactions cannot be decrypted or recovered on the COTS device after encryption.	To prevent the recovery of cleartext sensitive assets, such data needs to be stored so that it is not recoverable on the COTS device. The requirement is required to be enforced cryptographically through the use of keys that are securely removed from the COTS device immediately after encryption or using an approved asymmetric cryptography algorithm as described in *Appendix C: Minimum and Equivalent Key Sizes and Strengths for Approved Algorithms*. If symmetric cryptography is used, the COTS device cannot maintain sufficient information to recreate previous encryption keys after their removal. An example of symmetric cryptographic methods meeting this requirement would be use of a unique key per transaction key management, with each transaction key erased upon the completion of that transaction. In such an example, it is required that reconstruction of the encryption keys is not possible, except in the back-end system.

Security Requirements	Test Requirements	Guidance
1F-1.2 Payment transactions are only accepted in offline mode for a maximum period of 48 hours.	**1F-1.2.a** The tester must confirm through examination and observation that any further transaction processing is prevented when the COTS-based MPoC Software has been operating in offline mode for more than 48 hours.	Attackers may attempt to exploit offline processing to process multiple transactions that would otherwise be rejected if processed online. To reduce this risk, offline processing systems need to prevent further processing if they are prevented from connecting to the back-end system. The back-end payment-processing systems may be different to the back-end A&M processing systems. MPoC Software implementing offline payment processing need to achieve connectivity to the A&M back-end at least every 24 hours. The A&M connectivity requirement is different from this requirement and tested separately. This requirement is to halt payment processing once offline payment processing has been enabled for more than 48 hours.
1F-1.3 Data stored for offline transaction processing is not accessible to other applications.	**1F-1.3.a** The tester must confirm through examination and observation that the offline data is stored at a location not accessible to other applications.	Although offline data is required to be protected cryptographically, it also needs to be isolated from other applications as part of a defense-in-depth approach to protecting that data. In the context of this requirement “other applications” means applications that are not the MPoC Application. This requirement covers the storage of offline transaction data by the COTS-based MPoC Software. It does not prohibit the transmission of encrypted offline transaction data through other “calling” applications.
1F-1.4 Transactions currently underway during a transition to offline processing are either failed in a secure manner or managed in compliance with the requirements of this Section.	**1F-1.4.a** The tester must confirm through examination and observation that the transition to offline processing is managed securely, so that any transaction currently in process is either failed in a secure manner or is compliant to the requirements of this Section.	Online transactions may allow for operations that are not permitted as part of an offline transaction—e.g., online PIN capture is not permitted for offline payment processing. In cases where a transaction is currently underway during the transition to offline processing, it is important that the transaction is either failed in a secure manner, or the processing of that transaction is compliant to the requirements of this Section.

1F-2 Offline Monitoring

Solutions providing offline payment processing must implement attestation and monitoring functions on the COTS device that are able to operate securely without connectivity with the A&M back-end component. Although these requirements exist to support MPoC Solution where a system is disconnected from both the payment processing back-end systems, and the A&M back-end systems, it is noted that offline payment processing is a separate consideration and does not necessarily require that the back-end A&M systems are unavailable.

Security Requirements	Test Requirements	Guidance
Objective: The COTS-based MPoC Software can securely operate and attest its environment during periods where connection to the A&M back-end is lost.		
1F-2.1 The initiation of the offline mode is not available immediately after COTS device reboot. The COTS-based MPoC Software only works offline after being connected to the A&M back-end.	**1F-2.1.a** The tester must confirm through examination and observation that it is not possible to perform offline transactions if the A&M and payment-processing servers have not been contacted after a reboot of the COTS device.	The COTS-based MPoC Software cannot start in offline mode from a reboot state without first contacting the server. This helps mitigate attacks when the timer is manipulated by rebooting the COTS device or when the firmware is modified since the last contact with the A&M server. Validation of the local time as indicated by the COTS device during reconnections may be used as part of the A&M data. The ability to correctly indicate local time is important for the secure operation of the COTS-based MPoC Software.
1F-2.2 The COTS-based MPoC Software A&M component supports a separate attestation policy for offline operation.	**1F-2.2.a** The tester must confirm through examination that the COTS-based MPoC Software A&M component supports a separate attestation policy for offline operation. The COTS-based MPoC Software A&M component must support the following actions: • Confidentiality-protected logs of any checks that have a high chance of false positive results. • Impede the ability to transact until the A&M back-end is contacted for checks with high confidence of compromise (e.g., detection of specific hacking tools). • Local checks must include at least root detection, instrumentation detection, data-tampering checks, and local OS attestation if supported by the COTS platform.	It is expected that the COTS-based MPoC Software acts upon the results of the offline executed security checks to prevent fraud. The degree of certainty of the security checks needs to be defined in the A&M offline mode policy.

Security Requirements	Test Requirements	Guidance
1F-2.3 The A&M back-end implements controls to mitigate attacks attempting to delete the COTS-based MPoC Software during offline processing.	**1F-2.3.a** The tester must confirm through examination and observation that the A&M back-end implements controls to mitigate attacks attempting to delete the COTS-based MPoC Software during offline processing.	An attacker may attempt to delete the COTS-based MPoC Software from a COTS device during offline processing, so that any stored transactions are lost. Although this cannot be prevented, the A&M back-end is required to track offline use in attempts to identify such abuse. This requirement is intended to help mitigate attacks where COTS-based MPoC Software that has multiple offline transactions stored is deleted to remove those transactions. The requirement does not necessarily require the detection or response to any deletion of the COTS-based MPoC Software. For example, a system may flag merchants or COTS device instances where offline processing is enabled and the next appearance of that merchant or system is a new installation. This may not lead directly to disablement of that merchant but may lead to increased monitoring based on the attestation policy.
1F-2.4 The COTS-based MPoC Software disables payment acceptance after 24 hours without receiving a response from the A&M back-end allowing for the continued processing of transactions.	**1F-2.4.a** The tester must confirm through examination and observation that the COTS-based MPoC Software must disable processing after 24 hours without a response from the A&M back-end allowing for the continued processing of transactions.	It is not a requirement that the COTS-based MPoC Software supports offline payment processing; however, where this is supported, communications with the A&M back-end may be interrupted during this time. In such cases, it is acceptable for the disablement of online payment acceptance to be delayed for up to 24 hours.
1F-2.5 The results of the A&M security checks that are not dependent on the back-end are resistant to tampering to an attack rating of 25 points using the attack-costing framework in *Appendix B*. The verification/assessment of these security check results is not delayed until connection to the A&M back-end is reestablished.	**1F-2.5.a** The tester must perform testing on the COTS-based MPoC Software by triggering security checks in the COTS-based MPoC Software while in offline mode and verifying that they are active in offline mode, and that any response or reaction required by the attestation policy is enacted. **1F-2.5.b** The tester must perform testing on the COTS-based MPoC Software by attempting to extract card data and PIN data from the COTS-based MPoC Software when using the offline A&M configuration.	Security checks that do not depend on a back-end system to obtain a result—e.g., root detection, anti-tampering, etc.—need to be assessed locally while in offline mode.

Security Requirements	Test Requirements	Guidance
	1F-2.5.c The tester must provide a costing of this attack based on the method outlined in *Appendix B. Attack Costing Framework*. This requirement is passed if the most feasible attack cannot be costed for less than 25 points.	

Module 1G: MPoC Software Security Guidance and Integration

MPoC Software includes the COTS-based MPoC Software, which includes both payment and attestation functionality, and the back-end attestation and monitoring software.

This Module applies in all cases to MPoC Software that is to be separately listed as an MPoC Product. This Module additionally applies to solutions that separate the MPoC functionality from other parts of an MPoC Application, including those where the MPoC Solution Provider and the MPoC Application vendor are the same organization.

Monolithic MPoC Solutions are exempt from this Module as they do not use an MPoC SDK, and do not outsource the operation of their Attestation and Monitoring services.

1G-1 Security Guidance

The security guidance of a component is a document that defines the purpose and usage of the component, the security claims of the component, the component dependencies, and contains the required information to integrate and use the component in a secure manner. Different levels of guidance may be provided to different entities and parties in an overall MPoC Solution. For example, details on key management may be provided only to entities integrating and operating those aspects of the MPoC Software.

Security Requirements	Test Requirements	Guidance
Objective: The MPoC Software is provided with documentation that allows for secure integration and use within other MPoC Products.		
1G-1.1 The MPoC Software is provided with a security guidance document that describes how the MPoC SDK can be integrated with an MPoC Application. The security guidance document is made available to potential integrators and assessment laboratories.	**1G-1.1.a** The tester must confirm through examination that there is a security guidance document that is provided with the MPoC Software.	An MPoC Software needs to be integrated as expected by the MPoC Solution provider and the assessment laboratory to ensure that the validated security features are correctly implemented. The MPoC Solution provider is required to provide documentation about how this integration is achieved, and the laboratory needs to validate that this conforms to their expectations and testing setup.
	1G-1.1.b The tester must confirm that the security guidance document is made visible and readily accessible to all entities integrating the MPoC Software into their own systems or solutions, as well as assessment laboratories.	

Security Requirements	Test Requirements	Guidance
1G-1.2 The MPoC Software security guidance document defines the type of MPoC SDK that is implemented.	**1G-1.2.a** The tester must determine through examination, observation, and testing that the MPoC SDK provided as part of the MPoC Software is either an Isolating SDK, or a non-Isolating SDK. The tester may refer to the results from previous test items in making this determination.	An MPoC SDK may take one of two types. The first type is an Isolating MPoC SDK, which prevents the MPoC Application from accessing the memory or sensitive assets of the MPoC SDK. Any MPoC SDK that does not meet this requirement to isolate and protect its own memory and sensitive assets is considered a non-Isolating SDK.
	1G-1.2.b The tester must confirm through examination that the MPoC Software security guidance document correctly details the type of MPoC SDK implemented by the MPoC Software.	
1G-1.3 The MPoC Software is provided with a security guidance document that describes how the MPoC Software is to be operated by an Attestation and Monitoring Service provider.	**1G-1.3.a** The tester must confirm through examination that there is a security guidance document that is provided with the MPoC Software. This document must be made visible and readily accessible to all entities integrating the MPoC Software into their own systems or solutions.	An MPoC Software needs to be integrated as expected by the MPoC Solution provider and the assessment laboratory to ensure that the validated security features are correctly implemented. The MPoC Solution provider is required to provide documentation about how this integration is achieved, and the laboratory needs to validate that this conforms to their expectations and testing setup.
1G-1.4 The MPoC Software security guidance document defines an explicit MPoC SDK boundary.	**1G-1.4.a** The tester must confirm through examination that the security guidance document includes an explicit MPoC SDK boundary.	Clear documentation over the boundaries of the MPoC SDK is needed to scope the functionality of the MPoC Software properly and assess the evaluated components. The boundary has to include any hardware—e.g., PCI PTS POI, SE—that performs, or software that implements, any of the MPoC SDK-designated security functionalities. The MPoC SDK boundary needs to define how account data is input to the MPoC SDK, and how A&M and control signals pass between the MPoC SDK and other components of the Solution. Although interfaces such as the COTS-native NFC reader or touch screen are not part of the MPoC SDK boundary, it is required that they be considered part of any account data input methods into the MPoC SDK.
	1G-1.4.b The tester must confirm through examination that the MPoC SDK boundary, as defined, aligns with their understanding of the MPoC Software and includes details about how all account data and A&M data are input or output from that boundary.	

Security Requirements	Test Requirements	Guidance
1G-1.5 The MPoC Software security guidance document provides details about how the MPoC Software code and configuration settings can be updated securely.	**1G-1.5.a** The tester must confirm through examination that the required information is included and that it matches the understanding of the tester of the MPoC Solution.	Clear documentation about how updates to the MPoC Software and its configuration can be made is vital. Because the MPoC Software is designed for integration into other components of an MPoC Solution, updates may require specific processes or prerequisites. When an MPoC Software makes assumptions about the source or protections provided to updates, this needs to be clearly detailed in the security guidance documentation. Updates covered by this guidance includes data such as configuration for EMV kernels and updates for the MPoC Application to include the most recent versions of the MPoC SDK to cover new patches or updated SBC objects.
1G-1.6 The MPoC Software security guidance document provides details about how any software-protected cryptography implementations impact the frequency of MPoC Application updates.	**1G-1.6.a** The tester must confirm through examination that the MPoC Software security guidance documentation indicates how often any software-protected cryptography implementations must be updated. This time period must be no more than the minimum period included as part of the testing of the software-protected cryptography.	An important part of the security posture of any software-protected cryptography is the ability to provide regular updates. Any implementation of an MPoC SDK using a software-protected cryptography needs to ensure that it complies with this update process, which starts with providing this information in the MPoC Software security guidance documentation.
1G-1.7 The MPoC Software security guidance document contains detailed guidance for secure integration that includes configuration flags, usage of APIs, and expected security mechanisms to be applied, as applicable.	**1G-1.7.a** The tester must confirm through examination that the security guidance: • Matches the testers understanding of the MPoC Solution. • Is sufficient for secure integration of the MPoC SDK—i.e., it details how to integrate the MPoC SDK in a way that would enable the integrating MPoC Application to meet the relevant requirements of this standard.	The integrator is required to be aware of how to configure the MPoC Software properly for deployment and what security mechanisms are expected to be included during the development. For example, application permissions may significantly alter or impact the security of an MPoC Application, and assumptions made regarding the configuration of permissions during the MPoC Software evaluation, where changes to those permission could have such impact, need to be outlined in the MPoC Software guidance document to ensure assessment findings remain correct.

Security Requirements	Test Requirements	Guidance
1G-1.8 The MPoC Software security guidance document details how any secure channels that are able to be managed or configured by the MPoC Application are secured.	**1G-1.8.a** The tester must confirm through examination that the security guidance details: • How any configurations or implementations for the secure channels are to be performed in line with the MPoC requirements. • If a secure channel may be implemented by an MPoC Application, this is permitted only for use with third-party payment hosts.	Both Isolating and non-Isolating MPoC SDKs may allow for an MPoC Application to configure the secure channels that are implemented by the MPoC SDK. For example, a TLS-based secure channel may have certificates and/or URLs configured by the MPoC Application. Such configurations cannot include changing supported cipher suites to ones not permitted under the MPoC requirements. An MPoC SDK may allow for an MPoC Application to implement its own secure channel to a third-party payment host. Where this is possible, the MPoC Security Guidance is required to detail how the secure channel is to be implemented in line with MPoC requirements. Any MPoC Application implementing its own secure channels in this way is required to be validated to the requirements in Section 1A-1.5 during integration testing. Secure channels to any external devices or to the A&M back-end are required to be always managed by the MPoC SDK.
1G-1.9 The MPoC Software security guidance document indicates which COTS Platforms (including platform versions such as OS, TEE, and SE) and external devices (such as PCI PTS POI devices or non-PTS Approved MSRs) are supported.	**1G-1.9.a** The tester must confirm through examination that the required information is included and that it matches the understanding of the tester of the MPoC Solution.	The MPoC SDK is permitted to be used only on platforms that have been validated to provide the necessary security and functional aspects required. The integrator needs to limit the use of the MPoC SDK only to platforms where the MPoC SDK can be used securely. Sufficient granularity should be provided so that an integrator of the MPoC Software Product is able to know which platforms to target for their MPoC Application. This information must be kept current as changes in the supported COTS platforms occur over time.
1G-1.10 If the attestation needs an interaction from the MPoC Application, the MPoC Software security guidance document defines the scope, dependencies, and actors of an attestation policy that is used by the MPoC Solution.	**1G-1.10.a** The tester must confirm through examination that the required information is included and that it matches the understanding of the tester of the MPoC Solution.	MPoC SDK functionality may need interactions with the MPoC Application that are not functional, but which are needed for security. These interactions are required to be documented properly such that the integrator knows how to use the MPoC SDK properly. Interactions include specific API calls, configuration for periodicity and status, or other types of triggers.

Security Requirements	Test Requirements	Guidance
1G-1.11 The MPoC Software security guidance document provides details on the required key management processes and operations.	**1G-1.11.a** The tester must confirm through examination that the MPoC Software security guidance provides sufficient details on how to operate the key management required by the MPoC Software.	The MPoC Software is validated as implementing certain key management and encryption operations. However, all key management requires support from back-end systems, and it is required that the operational aspects of the key management are clearly defined in the MPoC Software security guidance document. Where different parties may be responsible for different aspects of key management, this is to be indicated in the document.
	1G-1.11.b The tester must confirm through examination that the guidance clearly outlines who is responsible for the management of each key management aspect of the MPoC Software.	
1G-1.12 Where vendor verification is to be used for the integration of their isolating SDK(s) procedures for the validation of MPoC Applications exist and are demonstrably in use.	**1G-1.12.a** The tester must confirm that: • The MPoC Software vendor is the sole party performing this testing (the testing is not outsourced to another organization). • The testing is performed only on isolating MPoC SDKs which are listed as part of that vendors MPoC Software product.	This requirement provides the scope for MPoC Software vendors to perform integration testing of an MPoC Application that integrates their own MPoC SDK. However, this requirement does not address the testing to be performed by the MPoC Software vendor themselves (referred to as Vendor Verification), but instead the testing to be performed by the MPoC Laboratory to validate the COTS-based MPoC Software vendor verification. Details on how MPoC Applications tested by MPoC Software vendors are managed within the MPoC Program are contained in the MPoC Program Guide. An MPoC Application that is integrating one or more MPoC SDKs must be tested to confirm that the integration has been performed securely and in line with the guidance provided for the MPoC SDKs it integrates. This testing may be performed by an MPoC Laboratory or by the vendor of the MPoC Software product that provides the SDK(s). Where testing is to be performed by the MPoC Software vendor, the processes and technologies used by that vendor must be confirmed to be sufficient. *(continued on next page)*
	1G-1.12.b The tester must confirm that a documented process for the validation of MPoC Applications integrating the vendors isolating SDK(s) exists. This procedure must cover: • Any and all isolating SDKs provided by the vendor. • That any and all SDKs which are not isolating SDKs provided by the vendor cannot be validated through this process. • All requirements of Domain 2A (at a minimum). • The process followed if the MPoC SDK integration is unable to be validated.	

Security Requirements	Test Requirements	Guidance
	1G-1.12.c The tester must review the vendor assessment process for at least two sample MPoC Applications for each isolating SDK provided by the vendor—one MPoC Application that contains an issue which would prevent validation to MPoC Domain 2A, and one which could be validated to Domain 2A. The tester must confirm: • The procedures have been followed as documented. • The validation results are correct based on the supplied MPoC Applications.	MPoC Applications that integrate MPoC SDKs from more than one MPoC Software vendor must be validated through an MPoC Laboratory. MPoC Applications that integrate an MPoC SDK that integrates another MPoC SDK must be validated either through laboratory testing, or through testing by the MPoC Software vendors of both MPoC SDKs (where both MPoC SDKs implement card-based payments). Where only one of the MPoC SDKs implements card-based payments, integration testing must be performed by the vendor of that MPoC SDK. The laboratory is expected to validate output from testing against listed MPoC Applications, where possible. In instances where a new MPoC Software product is under evaluation, and there are no currently listed MPoC Applications, test applications may be provided by the MPoC Software vendor. In this case, the laboratory is expected to make modifications to these test applications to ensure they can confirm the scope and correctness of the MPoC Software vendor testing process.

Domain 2: MPoC SDK Integration

The security requirements in this Domain apply to COTS-based MPoC Software that integrates a listed MPoC SDK or are developed as part of a monolithic solution. MPoC Applications may be developed by an MPoC Software vendor, an MPoC Solution provider, or by another party. However, the Entity responsible for the listing of an MPoC Product is also responsible for managing the assessment and listing of any MPoC Applications associated with that MPoC Product.

An MPoC Solution may have more than one MPoC Application listed as part of that MPoC Solution listing.

Further details on the different types of MPoC Applications can be found in Section: *MPoC SDKs and MPoC Applications*. Details on the applicability of the various Domains and requirements of this standard can be found in Section: *MPoC Domain and Section Applicability*. Details on the process and requirements for MPoC listings can be found in the MPoC Program Guide.

Module 2A: MPoC SDK Integration

This Module covers the integration of one or more MPoC SDKs, which are part of listed MPoC Software Products, into an MPoC Application or another MPoC SDK (where that integrating MPoC SDK does not provide additional card-payment features, or accesses cleartext sensitive data). Any one MPoC Application is not permitted to integrate more than two MPoC SDKs.

An MPoC Application that integrates both an Isolating and non-Isolating MPoC SDK is assessed against requirements as they apply to the integration of a non-Isolating MPoC SDK. An MPoC SDK that integrates another MPoC SDK where the integrating MPoC SDK provides additional card-payment features, such as accepting COTS-native NFC or PIN entry, or provides an interface to an external card reader, is required to be assessed against the relevant requirements of Domain 1.

Note: *This Module is applicable for MPoC Applications that integrate an MPoC SDK, regardless of the type of MPoC SDK used (Isolating SDK or non-Isolating SDK). It is also applicable to an MPoC SDK that integrates an Isolating SDK, but where the integrating MPoC SDK does not itself provide additional card-payment features or access cleartext sensitive data. Monolithic MPoC Applications are exempt from this Module.*

2A-1 Secure MPoC SDK Integration and Usage

An MPoC SDK, when used, must be securely and correctly integrated into an overall MPoC Solution that includes the integration into the MPoC Application distributed and used by merchants. This integration must follow the guidance provided by the MPoC Product vendor, as well as ensuring requirements of this standard are met for the integrated system. The testing in this Section cannot be performed through review of documentation alone, it is a requirement that the COTS-based MPoC Software is validated to these requirements. Automated systems may be used where appropriate.

Security Requirements	Test Requirements	Guidance
Objective: When an MPoC SDK is used, it is integrated into other COTS-based MPoC Software securely and correctly.		
2A-1.1 When an MPoC SDK is used, the MPoC SDK is part of a listed MPoC Product.	**2A-1.1.a** The tester must confirm through examination that the version of the MPoC SDK used is listed on the PCI website as part of an existing MPoC Product.	MPoC Products, including the MPoC Applications are required to be evaluated in their entirety before approval. When an MPoC SDK is relied upon for some aspect of compliance, that MPoC SDK needs to have been previously assessed and listed as part of an MPoC Product.
2A-1.2 The MPoC SDK is integrated with the COTS-based MPoC Software and other aspects of the MPoC Product in accordance with the MPoC Software security guidance.	**2A-1.2.a** The tester must confirm through examination and observation that the MPoC SDK has been integrated into the COTS-based MPoC Software in accordance with the security guidance for that MPoC SDK.	An MPoC SDK provides various security services and functions to assist with the overall compliance of the MPoC Product. To ensure correct and secure operation the MPoC SDK needs to be integrated and used as intended. An MPoC SDK may rely on the integrating COTS-based MPoC Software to provide a deployment configuration to be used securely. The MPoC Application vendor needs to follow the requirements from the MPoC SDK security guidance—e.g., on the use of configuration flags, APIs, and protection mechanisms to be applied to integrating COTS-based MPoC Software.
2A-1.3 The COTS-based MPoC Software integrating the MPoC SDK does not bypass, circumvent, reimplement, or modify any of the security or operational features provided by the MPoC SDK. All card-based payment functions are provided by the integrated MPoC SDK(s).	**2A-1.3.a** The tester must confirm through examination and observation that the integration into the COTS-based MPoC Software did not modify or reimplement MPoC SDK features that are within the MPoC SDK boundary.	An MPoC SDK is evaluated and approved based on the features it provides. Alteration or bypassing any of those features may impact the security of the MPoC SDK in unexpected ways. Therefore, any MPoC SDK needs to be used only as outlined in the provided security guidance documentation. For example, an MPoC Application that integrates an MPoC SDK that provides PIN entry cannot bypass this function to provide its own PIN entry or encryption methods.
	2A-1.3.b The tester must confirm through examination and observation that all card-based payment functions within the COTS-based MPoC Software are provided by the MPoC SDKs integrated by that application.	

Security Requirements	Test Requirements	Guidance
2A-1.4 The COTS-based MPoC Software that is integrating the MPoC SDK does not manage, process, or provide for the input of any sensitive assets, or the COTS-based MPoC Software is assessed to the requirements of Domain 1.	**2A-1.4.a** The tester must confirm through examination and observation that the COTS-based MPoC Software that is integrating the MPoC SDK does not manage, process, or provide for the input of sensitive assets such as cardholder PINs, cryptographic keys, or account data.	The COTS-based MPoC Software that integrates the MPoC SDK relies on the MPoC SDK to manage all sensitive assets such as cardholder PINs and account data. Where the COTS-based MPoC Software manages, processes, or provides input to any sensitive assets, the COTS-based MPoC Software is required to be assessed against Domain 1 of this standard. Assets that are encrypted by the MPoC Software (or by a system or entity external to the COTS device) using cryptographic methods and key management assessed under Domain 1 of this standard are considered suitably protected and not in scope of this requirement.
	2A-1.4.b Where the COTS-based MPoC Software that is integrating the MPoC SDK is found to manage, process, or allow for the input of sensitive assets, the tester must confirm that the COTS-based MPoC Software has been included in the scope of the Domain 1 assessment.	
2A-1.5 Attestation functions provided by the MPoC Software are securely and correctly integrated into the COTS-based MPoC Software.	**2A-1.5.a** The tester must confirm through examination and observation that the MPoC Software A&M functions are integrated into the COTS-based MPoC Software as required in the security guidance document of the MPoC SDK being integrated.	The MPoC Application is the method by which the MPoC SDK functions are delivered to the merchant. Compromise of an MPoC Application may prevent or alter the secure operation of the MPoC Solution, and so the security of the MPoC

Security Requirements	Test Requirements	Guidance
	2A-1.5.b The tester must test the A&M system by blocking the attestation from happening, either by tampering with the MPoC Application or the communication with the A&M back-end and attempt to perform payments and verify that it is mandatory to perform attestation as required by the A&M policy.	Application must be evaluated as part of the attestation and monitoring process. Attestation functions provided by the MPoC SDK may require specific integration steps or interactions with the MPoC Application. For example, specific triggers or data points to support the secure operation of the attestation functions may be required within the MPoC Application. These interactions are required to have been documented properly as part of the MPoC SDK validation. The process that this documentation outlines needs to be followed to ensure that the MPoC SDK provides the security features that are expected of it. Attestation needs to be an integral part of the MPoC SDK and MPoC Application flows. This means that initiating transactions, without having performed and passed attestation, needs to be prevented. This control may be enforced in the payment-processing environment—e.g., by requesting proof of successful attestation. The payment-processing environment is required to be aware when attestation is due, and not process data unless attestation has happened. It is required that the MPoC Application becomes unable to transact if communication between the MPoC Application and attestation API(s) is blocked. For offline use cases, this requirement still applies. The MPoC Application needs to perform local attestation during offline periods and needs to ensure successful A&M connection and validation prior to any further payment processing when coming back online.

Security Requirements	Test Requirements	Guidance
2A-1.6 The MPoC Application does not integrate more than two MPoC SDKs.	**2A-1.6.a** The tester must confirm through examination and observation that the MPoC Application does not integrate more than two MPoC SDKs.	Integration of multiple MPoC SDKs is permitted but increases the complexity and potential attack surface of the MPoC Application. To limit the risk posed by integration of multiple MPoC SDKs, no more than two may be integrated into a single MPoC Application.
2A-1.7 The COTS-based MPoC Software integrating the MPoC SDK does not implement, or allow for, the decryption of encrypted sensitive assets output by the MPoC SDK.	**2A-1.7.a** The tester must confirm through examination and observation that the COTS-based MPoC Software integrating the MPoC SDK does not implement or allow for the decryption of sensitive assets encrypted by the MPoC SDK.	During assessment to Domain 1, it is established that the MPoC SDK only outputs sensitive assets when that data is appropriately protected—e.g., through the use of encryption. It is important that the COTS-based MPoC Software that integrates that MPoC SDK does not attempt to bypass this security, for example by implementing decryption functions to recover the cleartext sensitive assets.
2A-1.8 The MPoC Application does not share assets between different MPoC SDKs.	**2A-1.8.a** Where more than one MPoC SDK is integrated into an MPoC Application, the tester must confirm through examination and observation that assets are not shared between the different MPoC SDKs.	An MPoC Application may integrate up to two MPoC SDKs. However, in cases where this is done the MPoC SDKs must continue to operate independently of each other, even if this leads to differences in user experience. Put another way, the existence or removal of one MPoC SDK should not impact the operation or security of another MPoC SDK integrated into a single MPoC Application. For the purposes of this requirement “assets" are considered to be those items defined as per the section “Security Objectives and Assets.”
2A-1.9 The MPoC Application is assigned only the privileges required for its secure operation.	**2A-1.9.a** The tester must confirm through examination that the MPoC Application is assigned only the privileges it requires for secure operation.	It is common for COTS Platforms to require that specific privileges are assigned to each application that executes on that platform. It is important that the MPoC Application is only assigned those privileges it requires for secure operation.

Security Requirements	Test Requirements	Guidance
2A-1.10 COTS-based MPoC Software which implement its own secure channels for connections to third-party payment hosts meet the requirements of Section 1A-5.	**2A-1.10.a** The tester must confirm through observation if the guidance of the MPoC SDK requires or allows for the COTS-based MPoC Software that is integrating the MPoC SDK to implement its own secure channels. If so, the tester must validate that the MPoC Application meets the requirements of Section 1A-1.5.	An MPoC SDK may allow for the integrating COTS-based MPoC Software to provide its own secure channels for connections to remote payment hosts. In such cases the guidance for the MPoC SDK is required to clearly outline that this is permitted, and how the COTS-based MPoC Software may securely implement the secure channel.
	2A-1.10.b Secure channels implemented by the COTS-based MPoC Software integrating the MPoC SDK are used only for connection to the payment host.	
2A-1.11 The COTS-based MPoC Software that integrates the MPoC SDK is either unable to access any memory and storage locations where MPoC assets may be processed or reside, or the requirements of Module 2B have been met.	**2A-1.11.a** If the tester is unable to confirm through examination and observation that the COTS-based MPoC Software that integrates the MPoC SDK is unable to access the memory and storage locations used to process or store MPoC assets, all further requirements in this Module must be assessed.	Even though a COTS-based MPoC Software integrating the MPoC SDK may not be specifically designed to access MPoC assets such as cardholder PINs, cryptographic keys, or account data, the implementation of the MPoC SDK, or COTS-based MPoC Software integrating the MPoC SDK may allow for access to the memory or storage areas that contain these assets. In these cases, compromise of the COTS-based MPoC Software integrating the MPoC SDK will result in exposure of these sensitive assets. It is not sufficient to reference that the MPoC SDK that is integrated is an Isolating SDK. The testing under this requirement is intended to validate that the integration has been performed sufficiently and in accordance with guidance, even in the case of an Isolating SDK.

Module 2B: MPoC Application Security

This Module covers all security mechanisms required of MPoC Applications that share memory with, or have access to the memory of, the COTS-based MPoC Software they are integrating. This Module will always apply to monolithic MPoC Applications, and MPoC Applications that integrate a non-Isolating MPoC SDK. MPoC Applications that integrate an Isolating MPoC SDK but do so in a way that compromises the isolation provided by that SDK, must also be assessed to the requirements of this Module.

The MPoC Software security guidance, and MPoC Software listing, details if an MPoC SDK is an Isolating SDK or non-Isolating SDK.

2B-1 MPoC Application Security

MPoC Applications that may be able to access assets of the COTS-based MPoC Software must be developed in line with security best practices for COTS device applications.

Security Requirements	Test Requirements	Guidance
Objective: MPoC Applications that could have access to assets are developed in line with security best practice.		
2B-1.1 All software in the MPoC Application is developed by an entity that either: • Meets the requirements of the PCI Secure Software Lifecycle (SLC) Qualified Software standard, or • Meets the requirements of Appendix D.	**2B-1.1.a** When the software in the MPoC Application is developed by a PCI Secure SLC-approved software vendor, the tester must confirm through examination that the entity is either listed on the PCI website and it is valid at the time of evaluation—e.g., the listing has not expired—or the tester must validate the entity against the Secure Software Lifecyle requirements.	A non-Isolating SDK has not been validated to provide sufficient protection to its assets. Compromise of an MPoC Application that integrates a non-Isolated SDK can therefore lead directly to the compromise of the MPoC SDK assets. Therefore, it is important that any MPoC Application that integrates a non-Isolated SDK is created using secure software development best practices. This will help reduce the potential for vulnerabilities in the MPoC Application, increasing the security of the overall MPoC Solution.
	2B-1.1.b Where the software is not developed by a PCI Secure SLC-validated software vendor, the tester must confirm through examination and observation that the MPoC Software vendor meets the requirements of Appendix D.	

Security Requirements	Test Requirements	Guidance
2B-1.2 The MPoC Application data and code are protected against tampering (modification), including at runtime, to an attack rating of 25 points using the attack-costing framework in *Appendix B*. Protections must include controls to mitigate attempts to perform rollback on the MPoC Application or COTS OS.	**2B-1.2.a** The tester must confirm through testing that the modification (including application rollback) is detected by the MPoC Application, and it is not possible to perform transactions. If the modification is detected, the tester must attempt to bypass—i.e., disable or remove—the tamper detection code.	The MPoC Application installable package needs to have its authenticity protected. This is normally a task that the MPoC Software cannot perform, as it has no visibility on the integration part of the MPoC Application. The MPoC Application binary code is required to be resistant against tampering. The MPoC Application is required to implement controls to prevent modification of the MPoC Application installed on the COTS device, including its configuration files, and binary code. The MPoC Solution needs to prevent older and possibly vulnerable versions of the MPoC Application from running. The version allowed to be used is controlled by the MPoC A&M back-end, and validation of this functionality is performed in Module 1C. A deprecated version is an older version not allowed to be operational in the field due to being too old or compromised.
	2B-1.2.b The tester must provide a costing of this attack based on the method outlined in *Appendix B. Attack Costing Framework*. This requirement is passed if the most feasible attack cannot be costed for less than 25 points.	

Domain 3: Attestation and Monitoring

These requirements cover the operational aspects of the A&M. If an A&M service provider wants to have its A&M Service listed independently from an MPoC Solution, as part of an MPoC Service, the A&M service provider is responsible to ensure that the requirements in this Domain are met. When an MPoC Solution provider is not using an MPoC Service that provides A&M functions, the MPoC Solution provider is responsible to ensure that the requirements in this Domain are met.

Module 3A: MPoC Software Security Guidance Compliance

This Module covers the deployment and configuration of a certified attestation and monitoring software into the back-end.

Note: *This Module is applicable for composite solutions only.*

3A-1 Deployment and Configuration of Back-end Systems

An MPoC Software product, when used, must be securely and correctly integrated into an MPoC Service or MPoC Solution. This integration must follow the guidance provided by the MPoC Software vendor, as well as ensuring requirements of this standard are met for the integrated system.

Security Requirements	Test Requirements	Guidance
Objective: MPoC Software back-end systems, when used, are securely and correctly deployed and configured.		
3A-1.1 When an MPoC Software product is used, the MPoC Software product is listed on the PCI SSC website.	**3A-1.1.a** The tester must confirm through examination and observation that the version of the MPoC Software used by the Attestation and Monitoring Service provider is PCI approved.	MPoC Solutions need to be evaluated in their entirety before listing. When an MPoC Software is relied upon for some aspect of compliance, that MPoC Software needs to have been previously assessed and approved.
3A-1.2 Back-end systems are deployed and configured in accordance with the MPoC Software security guidance.	**3A-1.2.a** The tester must confirm through examination and observation that the Attestation and Monitoring Service provider has deployed and configured the back-end system in accordance with MPoC Software security guidance.	The MPoC Software back-end system provides various security services and functions to assist with the overall compliance of the MPoC Solution. To ensure correct and secure operation, back-end attestation system and back-end monitoring system need to be deployed and configured as intended.

Security Requirements	Test Requirements	Guidance
3A-1.3 The Attestation and Monitoring Service provider does not bypass, circumvent, reimplement, or modify any of the security or operational features provided by MPoC Software.	**3A-1.3.a** The tester must confirm through examination and observation that the integration of the MPoC Software did not modify or reimplement MPoC Software features that are within the MPoC Software boundary.	An MPoC Software is evaluated and approved based on the features that it provides. Alteration or bypassing of any of those features may impact the security of the MPoC Software and the MPoC Solution in unexpected ways.
3A-1.4 The Attestation and Monitoring Service provider has processes in place to detect when their back-end systems require updates.	**3A-1.4.a** The tester must confirm through examination that the back-end A&Ms supports secure updates according to the MPoC Software security guidance document.	The MPoC Software may be developed by an entity different to that which deploys and operates the back-end attestation system and back-end monitoring system. Even where the same entity develops and operates the back-end systems, there may be separate teams or resources used for each. To ensure that back-end systems remain up to date, the attestation monitoring service provider needs to have processes in place to monitor published updates and releases. Although newly developed systems may not yet have a need to implement updates, the need to continually monitor for such need remains. It is also expected that any A&M service provider will need to implement updates frequently to maintain the currency and security posture of their systems, and therefore even newly developed systems will require updates within a short period of time.
	3A-1.4.b The tester must confirm through examination that the A&M service provider has a process to implement updates as required, and this process is in demonstrable use.	

Module 3B: Attestation and Monitoring

This Module contains requirements for the operation of the attestation & monitoring service.

3B-1 Attestation and Monitoring Policy

The attestation and monitoring system policy defines what data types are collected during the attestation process, and how this data is to be interpreted and managed through the attestation and monitoring components of the MPoC Software.

Security Requirements	Test Requirements	Guidance
Objective: The COTS devices used provide a secure and trustworthy execution environment.		
3B-1.1 A documented attestation policy exists and is demonstrably in use.	**3B-1.1.a** The tester must confirm through examination that an attestation policy exists and includes the following topics at a minimum: • Roles and responsibilities. • Data collected during attestation. • Risk rating of attestation data. • Follow-up actions and time frames when attestation data seem to suggest a tampered component. • References to specific procedures and/or work instructions.	The attestation policy outlines what the requirements are to ensure that any COTS device executing the MPoC Software provides a secure execution environment for the MPoC Software. The attestation policy differs from the MPoC Software guidance documents referenced in Requirement 3A-x in that it defines the appropriate responses and actions to be performed with respect to the output of the A&M system. A monolithic MPoC Solution may not include MPoC Software guidance, as it is developed and deployed entirely in-house. However, all attestation and monitoring systems will always have an attestation policy that defines how the A&M system operates, the roles and responsibilities of those using or on an escalation path, how risks are ranked, etc. Put another way, the requirements in 3A-x consider how the A&M system is configured and deployed, and the requirements in this Section consider how the A&M system is to be used and operated.
	3B-1.1.b The tester must confirm through examination that the attestation component is configured according to the attestation policy and attestation and monitoring (A&M) integration requirements for the MPoC Software.	
	3B-1.1.c The tester must confirm through examination, observation, and interview that the configuration is verified periodically.	

Security Requirements	Test Requirements	Guidance
3B-1.2 If the MPoC Software supports offline transactions, the documented attestation policy contains an explicit offline operation mode.	**3B-1.2.a** The tester must confirm through examination that an Offline Attestation Policy exists, and it includes at a minimum: • Checks are enabled when operating in offline mode. • The responses when a security check is triggered. • Requirements for escalation are noted when offline transaction processing violates the requirements of Section 1F.	Attestation and monitoring functions provided to MPoC Solutions that allow for offline transactions need to account for the potential operation of those systems while they are disconnected from the A&M back-end systems. Section 1F of this standard details requirements for the MPoC Software when operating in offline mode. This includes preventing offline acceptance for more than 48 hours, requiring online attestation to be performed prior to enablement of offline mode, as well as other requirements. It is important that the offline A&M policy outlines the escalation path required if any of these functions of the MPoC Software are found to have been somehow bypassed or violated.
	3B-1.2.b The tester must confirm through examination, observation, and interview that the configuration is verified periodically.	
3B-1.3 If offline operation is implemented, the offline attestation policy is demonstrably in use.	**3B-1.3.a** The tester must confirm through examination, observation, and interview that the offline provisions in the attestation policy are demonstrably in use for any system operating in offline mode.	
3B-1.4 Personnel involved in maintaining the operation of the A&M are appropriately skilled.	**3B-1.4.a** The tester must confirm through examination, observation, and interview that the personnel involved with the operation of the A&M are appropriately skilled. This must include personnel skilled with the security of the COTS platforms used in the baseline.	The A&Ms are a combination of automated and manual features. Although the A&M software may be developed and maintained by a third party (the MPoC Software vendor), the operation of this software will require personnel who are sufficiently skilled in the platforms used. As the security of COTS platforms is a rapidly evolving field, ongoing training or research is expected for the personnel operating the A&Ms.

3B-2 Monitoring

Attestation data must be analyzed for signs of potential attack or tampering attempts. This monitoring process is an integral part of a whole attestation and monitoring solution and must be performed in a secure and repeatable manner.

Security Requirements	Test Requirements	Guidance
Objective: Attestation data is analyzed, and signs of potential attack are consistently responded to in a way that maintains the security and integrity of the MPoC Solution.		
3B-2.1 There is an overview of the MPoC Solution being monitored.	**3B-2.1.a** The tester must confirm through examination that a list of all components that are included in the monitoring process exists and contains references to how these are integrated into the A&M. This is required to include at least: • The platforms supported and operated. • MPoC Application and MPoC SDK (if used). • Any attached card reading devices.	A clear and thorough understanding of the components used in the solution, and how they are being monitored, is vital. This includes the identification of both OS versions, and COTS device types, as the security posture of each individual platform may have an impact on the security of payment processing. For example, some platforms may implement logging of NFC, or touch data at the OS level, or may provide system applications with higher level privileges that could result in the compromise of account data. It is not a requirement that all platforms of concern are prevented from accepting payments, but such platforms may be required to undergo additional or enhanced A&M checks to verify their secure status.

Security Requirements	Test Requirements	Guidance
3B-2.2 A documented operational procedure for monitoring exists and is demonstrably in use.	**3B-2.2.a** The tester must confirm through examination that monitoring procedures exist and that they contain the necessary information. The monitoring procedure needs to include at a minimum the following topics: • Definition of possible events (warning alerts, errors, etc.) from the monitoring system. • How specific events are processed. • How undocumented, unexpected, and unknown events are handled. • When and how events are escalated and when the incident management process is initiated. • How it is ensured that personnel are qualified to handle events. • References to work instructions for the actual MPoC Software used.	This requirement ensures common understanding for those involved in the monitoring of the solution. The monitoring process may include both manual and automated aspects.
	3B-2.2.b The tester must confirm through observation and interview that the monitoring procedures are actually executed. The required evidence is an overview of events that have been managed. Information about events must include date, time, description, and follow-up action.	

Security Requirements	Test Requirements	Guidance
3B-2.3 The A&M results are made available to the payment processing back-end.	**3B-2.3.a** The tester must confirm through examination and observation that the A&M provides for an indication of A&M results to the payment processing back-end.	The A&M provides "health" checks of the COTS platform and MPoC Application to ensure the secure operation of the MPoC Solution. When the attestation and monitoring (A&M) checks indicate that there may be a problem with any MPoC Application or COTS platform, the A&M provides that information to the payment processing back-end so that the continued operation of that specific MPoC Application can be determined. In cases where there is sufficient information indicating that the MPoC Application is at risk of compromise, the payment processing back-end needs to be informed so that payment processing for that MPoC Application can be stopped. This requirement does not dictate how the attestation and monitoring (A&M) results are communicated but does require that the results are current and relevant for the purposes of determining risk to ongoing transaction processing (as attestation results may not be indicative of a clear pass/fail result, so risk-based decisions may be required). A&M results may be provided to the payment processing back-end by programmatic or operational methods.

Module 3C: Operational Security

This Module contains the requirements regarding the operational security of the A&M back-end systems.

3C-1 Operational Management

This Section covers the operational management of the A&M service.

Security Requirements	Test Requirements	Guidance
Objective: A&M assets processed in back-end environments are secured according to their protection types.		
3C-1.1 Where A&M systems are sufficiently isolated from any payment processing systems and Cardholder Data Environment the A&M environment complies with the relevant requirements defined in *Appendix A: Back-end Environment Security Requirements*	**3C-1.1.a** The tester must confirm through examination, observation, and interview that account data is not stored, processed, or transmitted through the back-end A&Ms, and that the A&Ms are sufficiently isolated from any such account data processing systems.	When the back-end A&Ms are isolated sufficiently from the cardholder data environment, which is used to process account data, it is not a requirement that the A&M systems also be compliant to PCI DSS. However, security controls are still required in the attestation and monitoring (A&M) environment, so assessment is required to be made against the requirements provided in the *Appendix A: Back-end Environment Security Requirements*. Although not PCI DSS, it is expected that assessment to *Appendix A* is performed with appropriately skilled and experienced individuals. On-site assessment is to be included where appropriate with consideration for any remote assessment guidance that may also exist. Appropriate separation may include isolation through networking controls to prevent the routing or access to account data from the attestation and monitoring (A&M) environment, or use of cryptographic controls—e.g., encryption of account data passed through the A&M environment, with no access to the decryption keys or functions from within that environment. Where sufficient isolation is not provided to the A&M environment, Requirement 4A-4.1 (PCI DSS validation) applies.
	3C-1.1.b The tester must confirm through examination, observation, and interview that the back-end A&Ms comply with the relevant requirements defined in *Appendix A: Back-end Environment Security Requirements*	

Security Requirements	Test Requirements	Guidance
3C-1.2 When the A&M service provider supports more than one MPoC Solution, the assets of the different MPoC Solution providers are segregated.	**3C-1.2.a** The tester must confirm through examination and observation that the assets of the different MPoC Solution providers are segregated when operated by a single A&M service provider.	The specifics of how any segregation is actually implemented is based on a risk assessment. This requirement does not preclude the pooling of anonymized attestation data obtained from different MPoC Solution providers for the purposes of increasing the efficacy of the A&M detection mechanisms. Penetration testing of the A&M service provider environment is required as part of PCI DSS, or *Appendix A* validation.
	3C-1.2.b The tester must confirm through examination and interview that the implemented segregation controls included in the scope of the A&M penetration testing process.	

Domain 4: MPoC Software Management

The security requirements and test requirements in this Domain cover the operational management of the software and key management aspects of the MPoC Software. This Domain must be implemented by at least one entity in the MPoC Solution, and there may be multiple entities who are required to comply with these requirements. For example, a monolithic MPoC Solution would require that the MPoC Solution provider is solely responsible for meeting the requirements of this Domain. Alternatively, an MPoC Solution may implement a separately listed MPoC Service provider that has been separately assessed to some or all of the requirements of this Domain.

Module 4A: Software Management

This Module contains the requirements for the secure operation and management of the software used in an MPoC Solution. This includes any COTS-based MPoC Software, A&M Software, and MPoC Application(s).

4A-1 COTS Software Distribution and Updates

Software that is to be installed and executed on the COTS device for an MPoC Solution needs to be compliant to this Section. This includes the MPoC Application that integrates the MPoC SDK, or the COTS-based MPoC Software itself if it is instantiated and distributed as a stand-alone MPoC Application. Unlike Domain 2 of this standard, which covers the development and software of the MPoC Application, this Section covers the operational aspects of distributing and maintaining an MPoC Application. This includes any ongoing key management requirements, as well as the requirements for the secure distribution of the MPoC Application(s).

These requirements do not dictate any specific type of distribution method for the MPoC Application, such as a dedicated OS Store, but they do outline the minimum requirements that any method used to distribute the MPoC Application must meet. This includes any initial loading of an MPoC Application to a COTS Platform as part of the manufacturing or pre-sales provisioning of that COTS platform.

Security Requirements	Test Requirements	Guidance
Objective: Software is securely provisioned to, and maintained on, the COTS device.		
4A-1.1 Information about how software is provisioned securely to the supported COTS devices exists.	**4A-1.1.a** The tester must confirm through examination that the information provided includes at a minimum: • The supported MPoC Application distribution methods. • How the MPoC Application distribution methods supported are protected. How access to MPoC Application distribution methods is protected (for the purposes of uploading or changing security sensitive settings). • How the data provisioned after the MPoC Application is installed and the purpose of that data. This includes executable files and configuration files.	The protection of the MPoC Application during the provisioning lifecycle stage is needed to prevent the distribution of a malicious MPoC Application. Documentation covering the MPoC Application distribution, and provisioning helps to identify vulnerabilities in this stage of the MPoC Software lifecycle. The MPoC Application distribution method may include the OS store of the COTS device, or an MDM solution. Regardless of what method is used, MPoC Applications are to be distributed in a secure manner, and in a way that protects their integrity and authenticity. Attackers may attempt to alter MPoC Application updates as they are transferred to a COTS device, or by altering the MPoC Application file in the store prior to distribution. In cases where the MPoC Application vendor is unable to verify or validate sufficient security in any particular distribution method, an alternative method of distribution is to be used. COTS platforms that are not able to provide any method for MPoC Application distribution that is sufficiently secure are not suitable for use.

Security Requirements	Test Requirements	Guidance
4A-1.2 The MPoC Application is installed and updated exclusively through defined COTS application distribution methods.	**4A-1.2.a** The tester must confirm through examination and observation that it is not possible to perform transactions on an MPoC Application loaded onto the COTS device through means other than the defined COTS application distribution methods.	The authenticity of the MPoC Application is a paramount concern in securing account data. 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 COTS device. In some OSs, it is possible to perform "side-loading" of applications—that is, install them separate to the normal application distribution methods and controls—at any time by configuring the COTS device to allow for this process. In other OSs, it may be necessary to establish a developer account or perform some other actions on the COTS device to allow for loading of applications outside the distribution store formally supported by the MPoC Solution. It is necessary to outline what processes are available to bypass the supported store(s) and confirm that these methods do not allow for the loading of an MPoC Application that will operate normally.
4A-1.3 The interface to each of the implemented MPoC Application distribution methods is protected.	**4A-1.3.a** The tester must confirm through examination that controls are implemented to prevent the compromise of any MPoC Application distribution channels used.	To distribute an MPoC Application, it must first be distributed to a COTS application distribution method. The upload access that MPoC Application vendors have with the COTS application distribution method is required to be protected to prevent tampering or replacement of the MPoC Application at this stage of the lifecycle. This can be done by limiting and securing the access that the MPoC Application development team and others have to the distribution channel.

Security Requirements	Test Requirements	Guidance
4A-1.4 The methods implemented to distribute the MPoC Application ensure the authenticity of the MPoC Application prior to initial execution.	**4A-1.4.a** The tester must confirm through examination and observation that all the methods supported for distribution of the MPoC Application provide authenticity prior to initial execution.	Only authentic MPoC Applications are permitted to be installed and operated as part of an MPoC Solution. The authenticity at installation time needs to be provided by the MPoC application defined distribution method and/or COTS platform, before the MPoC Application is executed for the first time following installation or updating. COTS platforms that do not allow for the use of a COTS application distribution method that can provide authenticity to the MPoC Application at installation time are not suitable for use with an MPoC Solution. The minimum requirements for the COTS platform baseline—including the need to support validation of a digital signature across MPoC Applications that are loaded onto the device—is covered in Requirement 4A-3.x. Some COTS platforms may support multiple signature types or allow for the use of self-signed certificates. The intent of this requirement is not to enforce that all signatures chain up to an authorized Certificate Authority, but that the COTS application distribution methods implemented provide for the ability to validate that the MPoC Application has not been maliciously modified since deployment to (and through) the COTS application distribution method.

Security Requirements	Test Requirements	Guidance
4A-1.5 The MPoC Application, or the distribution methods used for the MPoC Application, include methods to allow the merchant to validate the authenticity of the MPoC Application.	**4A-1.5.a** The tester must confirm through examination and observation that the MPoC Application, or the distribution methods used for the MPoC Application, implements methods to permit the merchant to validate the authenticity of the MPoC Application.	The defined COTS application distribution method may not guarantee that the application that is provided is genuine, instead simply providing confirmation that it did indeed come from the COTS application distribution method used. Therefore, the merchant needs to be able to separately validate the authenticity of the MPoC Application. Some COTS Platforms may support self-signed certificates, and it is not a requirement for a signature to chain up to an authorized Certificate Authority. Because of this, it is necessary that the merchant is able to separately validate that the MPoC Application they have downloaded is not just a valid application but is the MPoC Application they intended to download and use for payment acceptance. This requirement is intended to help secure MPoC Solutions against fake or “look-alike” MPoC Applications, created for the purpose of capturing payment data for malicious use. There are many ways in which this objective may be met—e.g., by using an out-of-band method (such as a website or phone number) to perform a challenge-response process with the MPoC Application, or by using the ability of the MPoC Application to communicate to an attached device, such as a PCI PTS POI, as confirmation it is indeed the valid MPoC Application. Bundling of the MPoC Application with the COTS platform or linking to the MPoC Application through the merchant enrollment process, so that it is infeasible for a user to incorrectly download a “fake” MPoC Application, may also be considered as a method for meeting this requirement.

Security Requirements	Test Requirements	Guidance
4A-1.6 The MPoC Application vendor implements a distribution method that is able to provide secure and timely updates of the MPoC SDK.	**4A-1.6.a** The tester must confirm through examination that the MPoC Application vendor implements a distribution method that is able to provide secure updates of the MPoC SDK according to the MPoC Software security guidance document. ***Note:*** *This requirement is concerned with the operational process of communicating and delivering updates, not with the technical process of maintaining and updating MPoC Software.*	The MPoC SDK may be developed by an entity different than that which develops and maintains the MPoC Application. Even if the same entity produces both the MPoC Software and the MPoC Application, there may be separate teams or resources used for each. In such scenarios, it is possible that the MPoC SDK has an update process and timing that are dislocated from that of the MPoC Application. The MPoC Software vendor is required to keep MPoC Application vendors informed of releases and changelogs as part of their validation and listing. MPoC Application vendors are required, on their part, to have procedures in place to coordinate the incorporation of new MPoC Software releases, including configuration settings.

4A-2 Key Management Operations

Encryption is relied upon as a foundational control in many MPoC security requirements. Security of the cryptographic keys, certificates, and systems used with these operations is of equal importance to the actual cryptographic algorithms used. This requirement applies to all operational aspects of key management, including those performed by MPoC Service providers, MPoC Solution providers, and MPoC Software vendors.

Security Requirements	Test Requirements	Guidance
Objective: Cryptographic keys and certificates are managed securely throughout their complete lifecycle.		
4A-2.1 Procedures to generate, distribute, revoke, and renew keys and certificates follow the MPoC Software security guidance and are demonstrably in use.	**4A-2.1.a** The tester must confirm through examination, observation, and interview, that procedures for the generation, revocation, and renewing of keys and certificates follow the MPoC Software security guidance and are demonstrably in use. The procedures must, at a minimum, include the following topics: • Relation to incident management processes. • Assessment of impact of key replacement (replacement of a root CA key will have more impact than a device specific key). • Communication to stakeholders. • Work instructions about how to actually revoke and renew specific keys and certificates.	The MPoC Software implements cryptography for various controls, and this implementation is assessed under the requirements of Domain 1. However, to ensure the security of the MPoC Solution, the ways in which the cryptographic systems are used and operated are also important. This includes any processes that involve the generation, revocation, renewal, and distribution of cryptographic keys. Operational aspects of an MPoC Solution are not permitted to implement their own cryptographic systems or protocols separately from those already implemented within the MPoC Software. This requirement covers the operational aspects of the key management already implemented in the MPoC Software design. The expectation is that no additional or altered key management has been implemented during implementation, and that the operational aspects of the existing key management are performed in an expected, secure, and compliant manner.

Security Requirements	Test Requirements	Guidance
4A-2.2 Secret or private cryptographic keys used for the security of the implementation in the back-end environments, which are not related to PIN security, are one of the following: • Protected through use of HSMs compliant to FIPS140-2/3 level 3, or PCI HSM requirements. • Protected through use of HSMs compliant to FIPS 140-2/3 level 2 and deployed in a Controlled Environment (as per the definition in ISO13491). • Unique per session and forward secret.	**4A-2.2.a** The tester must confirm through examination and observation that cleartext secret or private cryptographic keys in the back-end environments that are used for the security of the implementation and are not related to PIN security, are: • Stored in HSMs compliant to FIPS140-2/3 level 3 (or above), or PCI HSM requirements. • Stored in HSMs compliant to FIPS140-2/3 level 2 and deployed in a Controlled Environment (as per the definition in ISO13491). **4A-2.2.b** The tester must confirm through examination and observation that cleartext secret or private cryptographic keys that are related to PIN security, and any other cryptographic keys that are not unique per session and forward secret, are never exposed outside of a HSM in plaintext (where these keys are used for the security of the implementation).	Cryptographic keys need to be protected to prevent unauthorized or unnecessary access that could result in the exposure of encrypted data. Secret or private cryptographic keys need to be stored in an encrypted form or within an SCD, such as an HSM. This requirement applies to all keys used to encrypt account data (excluding PINs), as well as cryptographic keys used to secure attestation data. The security and management of PIN security related keys is assessed under the PCI PIN standard. The requirement for HSM use does not apply to cryptographic keys used for the establishment and security of secure channels, such as TLS keys. Cryptographic keys used for signing MPoC Applications, or that are used for creating software protected cryptography implementations, are also out of scope of this requirement. The two test requirements cover two aspects of the requirement—storage (when a key is not being actively used), and exposure during operation (which may include when a key is being used).
4A-2.3 Work instructions to operate HSMs exist and are demonstrably in use for each type of HSM used.	**4A-2.3.a** The tester must confirm through examination that HSM operational work instructions exist and contain the following topics at a minimum: • Confirmation of HSM security and operational state • Authentication/Authorization • Key generation • Key import • Key export • Key deletion • Audit log review	It is vital that HSM's are operated as prescribed by the HSM supplier. Documented work instructions help to prevent human failures. Keys that are required to be protected using HSMs do not include cryptographic keys used for secure channels, such as TLS keys. Application-level keys are included in scope and need to be protected by HSMs. Confirmation of the security and operational state of the HSMs may include physical observation or examination by the MPoC Service provider, or remote cryptographic means where physical observation is not possible—e.g., in Cloud HSM environments.

Security Requirements	Test Requirements	Guidance
4A-2.4 If the back-end systems include the use of cloud-based HSM's, the cryptographic keys are managed by relevant MPoC entity, and not accessible to the Cloud HSM provider.	**4A-2.4.a** The tester must confirm through examination and interview that cryptographic keys are generated and managed by the Cloud HSM user, and not accessible to the Cloud HSM service provider. **4A-2.4.b** The tester must confirm through examination, observation, and interview that cryptographic keys used in the back-end systems are not exposed outside of any cloud HSMs used in the solution.	Cloud service providers offer different ways to manage HSM keys. In general, three options are supported: • Keys are generated and managed by the Cloud service provider. Depending on the HSM type, this could even mean that HSM keys are shared with other customers. • Keys are generated and managed by the Cloud HSM user, but within the cloud environment. The keys are not shared with other customers, but security depends on the Cloud service provider's configuration. • Keys are generated by the Cloud HSM user in their own premises and the transferred securely to the Cloud service provider. Some Cloud HSM systems use the HSM only for storage of keys and allow for export of keys for cryptographic operations. These types of Cloud HSM systems are unsuitable for use with MPoC Solution. Keys that are required to be protected using HSMs do not include cryptographic keys used for secure channels, such as TLS keys. Application-level keys, including those keys used to encrypt A&M data sent to the back-end A&M environment, are included in scope and need to be protected by HSMs.
4A-2.5 Operational key management of secret and private cryptographic keys ensure the confidentiality of the keys throughout their lifecycle.	**4A-2.5.a** The tester must confirm through examination, observation, and interview that secret or private keys in the solution are maintained in one of the permitted forms throughout their lifecycle. Permitted key forms are: • Encrypted by a key encryption key of equal or greater strength (these key encryption keys need to satisfy this requirement). • Stored within an SCD. • Managed as two or more full-length components. • Managed as an M-of-N secret-sharing scheme.	Cryptographic keys are required to be protected against unauthorized disclosure. This includes ensuring keys are always managed in one of the "approved forms" as listed in this test requirement. This requirement applies to all keys used to encrypt PINs and account data, as well as cryptographic keys used to secure attestation data. The requirement for HSM use does not apply to cryptographic keys used for the establishment and security of secure channels, such as TLS keys. Cryptographic keys used for signing MPoC Applications, or that are used for creating software protected cryptography implementations, are also out of scope of this requirement.

Security Requirements	Test Requirements	Guidance
4A-2.6 Operational management of cryptographic keys ensure the integrity and authenticity of the keys throughout their lifecycle.	**4A-2.6.a** The tester must confirm through examination, observation, and interview that all keys in the solution have their integrity and authenticity protected.	Cryptographic keys are to be protected against unauthorized or unintentional change. For example, this can be achieved by: • Implementing cryptographic controls such as key blocks. Key blocks may be implemented entirely in the back-end systems, or in concert with the COTS-based MPoC Software (refer requirements in 1A-3.x and 1A-4.x). • Implementing key check value or key fingerprint verification processes. • Managing public keys in certificates.
4A-2.7 Management of secret and private cryptographic keys implement the principles of dual control and split knowledge.	**4A-2.7.a** The tester must confirm through examination, observation, and interview that dual control and split knowledge are implemented in the organization to manage secret and private cryptographic keys.	Split knowledge requires that no individual has any information about any part, even a single bit, of the cleartext key. Dual control requires at least two individuals to perform a process, as it is more difficult to establish a breach of process or information when multiple entities are required to conspire to misuse. There are several ways to implement dual control and split knowledge using logical mechanisms, physical mechanisms, or both. This requirement does not enforce the need for manual processes for keys that are instead always secured within an SCD. Depending on the number of cleartext components there may be different groups of key custodians, although it is required that there always be more than one. No one individual can ever have access to enough key components to reconstruct any part of the cryptographic key. For key custodians to be free from undue influence in discharging their custodial duties, different key custodians who are able to form the necessary threshold to create a key are not permitted to directly report to the same individual, except for organizations of insufficient size. Key custodians need to have regular security awareness training.

Security Requirements	Test Requirements	Guidance
4A-2.8 Mechanisms are in place to identify expired, invalid, and/or revoked certificates, and to prevent continued processing using certificates that have expired or been revoked.	**4A-2.8.a** The tester must confirm through examination and observation that a revocation mechanism exists and is effective in preventing the use of revoked certificates.	Expired certificates can introduce unacceptable risk to the solution. Payment functions are required to be halted when certificates that are relied upon have expired or are otherwise detected as no longer valid. Expired certificates may be an indication of a malicious user acting as an imposter of a legitimate organization or process who is phishing for sensitive information. Many security incidents are caused by expired or revoked certificates. While understanding of the keys used is important and collected in the key/certificate tables, it is equally important to have procedures to act upon if any key expires or is suspected of being compromised. The requirement is applicable to all components of the solution and includes only the certificates upon which the solution relies for security purposes.

4A-3 COTS Baseline and Vulnerability Management

These requirements cover penetration testing and vulnerability management for MPoC A&M service providers, and MPoC Solution providers.

Security Requirements	Test Requirements	Guidance
Objective: The security of systems relied upon by the MPoC Product is sufficient and maintained over time.		
4A-3.1 A penetration test has been performed on the interfaces between the COTS-based MPoC Software and back-end environments (e.g., A&M, payment processing and/or remote kernel) prior to the validation and listing of an MPoC Solution or A&M service provider, and at least once per year thereafter.	**4A-3.1.a** The tester must confirm through examination that a penetration test has been performed on the interfaces between the COTS-based MPoC Software in a suitable test application, and back-end environments prior to initial deployment, and at least annually thereafter. Penetration testing reports must be examined to confirm that the scope covers all aspects of the MPoC Solution, or A&Ms, and that any major vulnerabilities found during the penetration testing have been remediated.	All back-end entry points meant to parse and process data received from the MPoC Software are required to have undergone penetration testing. This includes the payment and PIN processing back-ends, as well as any cloud/remote kernel systems used, where these are part of the MPoC Solution. This penetration test differs from the test required in 1A-1.3 in that it is to be performed in the context of a malicious instance of COTS-based MPoC Software in a suitable test application, using a valid and authenticated secure channel to back-end systems. The testing is intended not to re-assess the security of the MPoC Software, but to confirm that the MPoC Software has been securely integrated into the other systems and software of the overall MPoC Solution. Because this is an operational compliance requirement, it is expected that source code and other details of the logical aspects of the MPoC Solution may not be available for this penetration test. Penetration testing and vulnerability management processes are expected to be part of any secure software lifecycle process. This requirement confirms the scope and efficacy of the penetration testing as it is applied to a complete MPoC Solution or an A&M service provider, as it has integrated and operates the MPoC Software. A penetration test is considered to be a clearly scoped and deliberately engaged activity performed on behalf of the MPoC Solution provider, or A&M service provider. This differs from the ad hoc testing that may be performed as part of a vulnerability management or flaw reporting program.

Security Requirements	Test Requirements	Guidance
4A-3.2 A security-flaw-reporting program is implemented to encourage the finding and reporting of vulnerabilities by internal and external entities.	**4A-3.2.a** The tester must confirm through examination that a vulnerability-reporting program exists for the system, and there is evidence of accepting and remediating security vulnerabilities found through this program.	Penetration tests need to be performed by suitably skilled resources and may be performed by resources internal to the entity if such resources exist. When penetration testing is performed by internal resources, the people performing the testing need to be separate from those who perform any operation, development, or integration work. Skills expected from the resources used for penetration testing include an understanding of EMV protocols and payment processing, skills and experience with mobile security and communications protocols, and a clear history of penetration testing experience. Results from annual penetration testing may not exist for newly deployed MPoC Solutions or A&M service providers but need to be provided for any review performed after the first year after their initial validation. However, an initial penetration testing report is required to be available prior to validation and listing of the MPoC Solution or A&M service provider. Penetration testing may be performed by the same entity that performs the MPoC evaluation, but the MPoC evaluation itself cannot be considered a penetration test to meet this requirement. A separate testing and reporting process must be implemented for this penetration test.
	4A-3.2.b For any vulnerabilities reported through the security flaw reporting program the tester must confirm through examination that any such vulnerabilities are processed through the vendor risk-and-update process and patched accordingly.	

Security Requirements	Test Requirements	Guidance
4A-3.3 A COTS platform baseline exists.	**4A-3.3.a** The tester must confirm through examination and observation that a COTS platform baseline exists and is validated by the A&M.	The COTS platform baseline outlines the COTS platforms on which the MPoC Application may execute and accept payment transactions. The establishment and maintenance of this baseline must include the minimum set of features required of the MPoC Software, both those outlined in this standard and those self-imposed by the MPoC Solution itself. The baseline may be instantiated in various ways and is expected to include use of both whitelist and specific block-list elements. For example, a baseline may include accommodation for COTS devices with a specific OS version, except for those COTS devices that also have certain undesirable features (such as NFC logging, or insecure OS modifications). The inclusion of any COTS device executing the MPoC Software must be validated by the A&M.

Security Requirements	Test Requirements	Guidance
4A-3.4 A procedure for managing the COTS platform baseline exists and is demonstrably in use.	**4A-3.4.a** The tester must confirm through examination that the COTS platform baseline procedure exists. The tester must examine the procedure to verify that it contains at a minimum: • Roles and Responsibilities—e.g., who is allowed to update the COTS platform baseline. • The acceptance process and criteria for adding COTS platforms to the COTS platform baseline (see next requirement). • How COTS platforms are added to the COTS platform baseline. • How decisions are made to remove previously acceptable COTS platforms from the COTS platform baseline. This must include a reference to the vulnerability management process. • How affected entities are informed of changes to the COTS platform baseline that will affect them.	The COTS platform baseline is not static and will change over time. Therefore, a procedure is required be in place to manage the existing baseline with respect to risks to the solution. There needs to be a clear process for accepting platforms into the COTS platform baseline that ensures the MPoC Application can execute and perform its functions as intended and in a secure manner. MPoC does not differentiate operating systems as supported or unsupported (by the OS vendor). Instead, MPoC requires that all platforms are validated and secured regardless of any ongoing support, or end of life of that OS. COTS-based MPoC Software that executes entirely outside of the REE of the COTS device, may allow for rooted and jail-broken devices to be included in the COTS platform baseline. In such cases, it must be confirmed that the COTS-based MPoC Software does not allow for sensitive assets, such as account data or cryptographic keys, to be exposed in, passed through, or obtained from the REE.

Security Requirements	Test Requirements	Guidance
	4A-3.4.b The tester must confirm through examination and observation that the COTS platform baseline includes: • The COTS platform's that are supported. • Each supported COTS platform provides at a minimum: – An enforcing mandatory access control framework. – A trusted boot mechanism that validates the OS authenticity. – Validation of an application cryptographic signature upon installation of applications. – Isolation of the interfaces and memory spaces used by different applications.	
	4A-3.4.c The tester must confirm through examination and observation that, for any COTS-based MPoC Software that executes in the REE, COTS operating modes that may impact security are not part of the COTS platform baseline.	
	4A-3.4.d The tester must confirm through observation and interview that the COTS platform baseline procedure is actually executed. The required evidence is an overview of changes to the COTS platform baseline. This must include a description of the changes, related change management tickets, and communication to affected merchants.	

Security Requirements	Test Requirements	Guidance
4A-3.5 The baseline COTS OS is regularly and frequently reviewed for vulnerabilities.	**4A-3.5.a** The tester must confirm through examination, observation, and interview that vulnerabilities in the COTS OS are analyzed periodically and assessed against known and unknown vulnerabilities.	It is intended that the vendor identifies the vulnerabilities that affect the system baseline. It is expected that not all platform vulnerabilities will result in risk to MPoC Solutions, but a process to determine if this is the case needs to be implemented. It is not sufficient for vulnerabilities to be dismissed without appropriate research and testing. Not all vulnerabilities that affect platforms within the COTS baseline require specific mitigations. Vulnerabilities that are not directly exploitable or cannot result in exposure or risk to the MPoC Solution assets may be considered and mitigations deferred until increased risk is noted. Security review of the COTS OS requires both threat detection and mitigation, as well as vulnerability detection and mitigation. This may include detailed review of each CVE for all supported platforms (in addition to other measures to identify unknown vulnerabilities), or it may instead be implemented through a combination of focused review of known vulnerabilities and robust periodic testing. For example, use of regular and focused penetration testing by mobile security experts instead of review of each individual CVE may be sufficient if (i) the testing is able to show the efficacy of the security mechanisms, (ii) there remains a continuous monitoring of the security landscape in between penetration tests. Re-assessment of existing vulnerabilities may be required as more details concerning the vulnerability are provided. For example, a vulnerability may be announced initially with little public detail, but as more details are made public the extent and impact of the vulnerability may become clearer over time.
	4A-3.5.b The tester must confirm through examination, observation, and interview that the identified vulnerabilities are mitigated through updates to the MPoC Application or A&M, in the process of having mitigations implemented, or are identified as having no security impact. The required evidence is: • Recent overview of assessed vulnerabilities including verdict. • List of mitigated vulnerabilities.	
	4A-3.5.c The tester must confirm through examination, observation, and interview that the frequency of the vulnerability review process is based on an informed risk decision and is no greater than every 3 months.	

Security Requirements	Test Requirements	Guidance
	4A-3.5.d Where the baseline includes devices or operating system versions which are not otherwise validated through other security testing, the tester must confirm through examination that these devices or operating system version are included in scope of the annual penetration testing.	MPoC allows for the use of devices and platforms which may not have otherwise undergone rigorous testing through other security validation methods. In such cases these items in the baseline are required to be included in the annual penetration testing. For example, a device may be designed for use in an enterprise environment and therefore not undergo the testing normally required by the Operating System vendor. In such circumstances, these platform and operating system versions are to be included in scope for the annual penetration test.

4A-4 Security of Back-end Systems

Back-end environments used as part of an MPoC Solution are secure and maintained in compliance with relevant standards or requirements.

Security Requirements	Test Requirements	Guidance
Objective: Back-end environments are implemented and operated securely and in ways that maintain compliance to other applicable standards.		
4A-4.1 Environments that store, process, or transmit account data comply with the requirements of PCI DSS (including environments implementing remote kernels).	**4A-4.1.a** The tester must confirm through examination that there exists a valid Attestation of Compliance (AOC) outlining compliance of any environment within the MPoC Solution that stores, processes, or transmits account data with the PCI DSS requirements.	Environments that are PCI DSS compliant demonstrate that the minimum set of industry-expected security controls have been applied to that environment, which reduces risk compared to environments that do not apply security controls. This includes any payment processing back-ends directly implemented by the MPoC Solution, or where there is the ability for the MPoC Solution systems to have access to cleartext account data or the cryptographic keys that can be used to decrypt any encrypted account data.
4A-4.2 Environments performing PIN processing, or that manage PIN related cryptographic keys, comply with the requirements of PCI PIN.	**4A-4.2.a** The tester must confirm through examination that a valid AOC outlining compliance of any PCI PIN-processing environment included in the MPoC Solution exists and is current and up to date with the features provided.	Environments that are PCI PIN compliant demonstrate that the minimum set of industry-expected security controls have been applied to that environment, which reduces risk compared to environments that do not apply security controls. Specific non-compliances raised as part of PCI PIN validation, due to the use of an MPoC Solution, need to be considered when assessing this requirement.
	4A-4.2.b The tester must confirm through examination that the key-loading facilities used for any PCI PTS devices implemented in the MPoC Solution are included in the PCI PIN compliance scope.	
4A-4.3 Environments that manage account data related cryptographic keys, comply with the requirements of Section 4A-2.	**4A-4.3.a** The tester must confirm through examination that any environments which manage account data related cryptographic keys comply with the requirements of Section 4A-2.	Cryptography is only as secure as the management of the keys used, and therefore it is essential that strong key management controls are applied to cryptographic systems used to secure MPoC implementations.

Security Requirements	Test Requirements	Guidance
4A-4.4 The A&M environment complies with the requirements in *Domain 3* of this standard.	**4A-4.4.a** The tester must confirm either that the A&M environment complies with the requirements of *Domain 3* of this standard or that the A&M service provider is listed on the PCI website as an approved A&M service provider.	The A&M environment may be operated by a service provider who already has been assessed and is listed as compliant to these requirements. Otherwise, the requirements in *Domain 3: Attestation and Monitoring* apply.

Domain 5: MPoC Solution

The security requirements and test requirements in this Domain cover the operational management of the complete MPoC Solution. This includes the validation of compliance for the payment-processing environment, the PIN-processing environment, the MPoC attestation and monitoring environment, and any other systems such as the split kernel environment.

Where the payment-processing and PIN processing are not performed by the MPoC Solution Provider those aspects may be considered out of scope of the assessment. Only those aspects that are included–e.g., a payment switch operated by the MPoC Solution provider–are to be included in the assessment of this Domain. However, it is not possible for an MPoC Solution that allows for PIN entry on the COTS device to never include validation of a PIN processing environment. Equally, it is also expected that all MPoC Solutions will include decryption or key management environments for account data sent from the MPoC Applications that are deployed as part of the MPoC Solution.

Although it is not necessary for the MPoC Solution provider to develop, implement, or operate all of these systems, it is the responsibility of the MPoC Solution provider to ensure that the relevant requirements for each are met. This may be achieved through a monolithic solution, where the MPoC Solution provider is responsible for the assessment and compliance of all aspects of the overall MPoC Solution, or through a composite solution where the MPoC Solution provider utilizes separately assessed and listed MPoC Products. In all cases, the MPoC Solution provider remains responsible for the compliance and security of the entire MPoC Solution.

Module 5A: Third-Party Management

This Module contains the requirements regarding the management of third parties.

5A-1 Merchant Identification and Communication

Initial provisioning of the MPoC Application includes providing for the unique identification of the merchant. Ensuring that the merchant is informed of changes or requirements of the MPoC Solution as the solution changes over time is vital to the security of the solution.

Security Requirements	Test Requirements	Guidance
Objective: The merchant is onboarded securely and kept up to date with relevant information in a timely manner.		
5A-1.1 The process of onboarding new merchants is documented. At a minimum, the process must describe how merchant identification is performed.	**5A-1.1.a** The tester must confirm through examination that the onboarding process is documented.	Identification is an important security measure against financial fraud to verify the identity, suitability, and risks involved with maintaining a business relationship with the merchant. This requirement is not intended to cover the legal or financial process of merchant on-boarding, but to ensure merchants are uniquely identified to help facilitate the A&M systems and any communications required.
5A-1.2 The merchant-onboarding process includes the provisioning of a unique merchant identifier, and the provisioning process of this unique merchant identifier is documented.	**5A-1.2.a** The tester must confirm through examination that the provisioning process is documented and describes how the unique merchant identifier is securely created and provided to the merchant. The tester must confirm that the required controls are implemented.	This requirement is primarily concerned with the data used to uniquely identify the merchant to the payment system. It does not require that the MPoC Application must implement a per-user password or other such identification system. However, if any passwords or other credentials are implemented, they must be transmitted and processed securely. Fraudulent activities often start with unauthorized access to credentials. The provisioning process is a particularly vulnerable process in the solutions. Well-known measures to improve the security of this process use separate channels and special PIN/password letters, and do not send user IDs and passwords together.

Security Requirements	Test Requirements	Guidance
5A-1.3 A communication plan exists, that includes how and when information is shared with merchants (or entities managing systems on behalf of merchants) and detail the information and trigger points that result in such communication.	**5A-1.3.a** The tester must confirm through examination that a communication plan exists and contains the following information at a minimum: • Changes to the system baseline. • Changes to any user manual including at a minimum: – How the MPoC Application must be installed and provided with unique (configuration) data. – How to use the MPoC Application securely—e.g., privacy during the customer PIN entry process. – Any security responsibilities that belong to the merchant when using the Solution. – User information about peripheral such as non-PTS approved MSR or PCI PTS POI if used in the Solution. – How to contact the MPoC Solution provider. • Planned maintenance downtime. • Provisioned credentials. • Information about potential compromise/security incidents. • End User License Agreement (EULA).	It is important that the merchant receives relevant information in a timely manner, and this process cannot be performed ad hoc when issues arise. A merchant communication plan ensures that the processes for how and when to communicate to the merchant base are detailed, so all parties are aware of their obligations. For example, a merchant would need to be notified if their COTS device was becoming too old to be accommodated in the MPoC Product baseline, or if there was information provided by the A&M that indicated a potential compromise. This communication does not need to be passed through, or be part of, the acquiring relationship with the merchant. For example, communications could be provided through the MPoC Application directly. Validation of this requirement may involve examination of previously issued merchant communications, or demonstrations of how an MPoC Application may display communications to merchants or agents managing systems on behalf of merchants.
	5A-1.3.b The tester must confirm through examination that the communication plan is demonstrably in use.	

5A-2 Support for Multiple Entities in the Solution

The MPoC Solution is comprised of multiple different components, such as the MPoC Software, MPoC Application, the A&M back-end, and the payment-processing back-ends. There must be a clear process for understanding and defining the requirements and roles and responsibility of each entity involved.

Security Requirements	Test Requirements	Guidance
Objective: Formal processes exist to define and align the interaction and roles of the different entities in the MPoC Solution.		
5A-2.1 Documentation exists that describes how the operational processes are aligned between the different entities in the MPoC Solution.	**5A-2.1.a** The tester must confirm through examination that documentation on cooperation exists and that it contains the required information. Documentation must contain the following at a minimum: • Contact information of all entities involved. • Agreements on incident management processes. • Agreements on escalation processes.	The Solution might be operationally managed by different entities. This often leads to problems in case of incidents covering more than one entity. Agreements on these kinds of topics help to solve potential incidents efficiently. For example, in the case where a specific COTS Platform is removed from the A&M baseline due to security concerns, it must be clear how this process of removal is to be handled between the various parties who comprise the MPoC Solution.
5A-2.2 The MPoC Applications deployed by the MPoC Solution are supported by the Attestation and Monitoring systems implemented.	**5A-2.2.a** The tester must confirm through examination that the MPoC Applications deployed by the MPoC Solution are supported by Attestation and Monitoring Services (or internal systems) implemented.	An MPoC Solution may be comprised of multiple MPoC Software products, or a combination of MPoC Software Products and monolithic MPoC Applications. To ensure that the security of the MPoC Solution is maintained, it is important that the MPoC Applications deployed are supported by the Attestation and Monitoring Services (or internal systems) used by the MPoC Solution. For example, it would not be acceptable for an MPoC Solution to use the MPoC SDK from MPoC Software Product A, but not use an Attestation and Monitoring Service (or internal system) that also supports that same MPoC Software Product.

Appendix A Back-end Environment Security Requirements

Recognizing that back-end attestation systems that support the MPoC Solution may not be in scope of PCI DSS, this appendix defines the minimum requirements to ensure fundamental security of the back-end monitoring system and back-end attestation component.

This appendix is referenced through test items outlined previously in the MPoC standard, where security assurance of those systems is required, but assessment to PCI DSS is not otherwise in scope. Where assessment against the requirements in this appendix is to be performed, the MPoC laboratory must confirm that all relevant systems and personnel are included. This may include subsets of all possible systems or personnel if sufficient clarity on the systems and operations of the environment can be determined. The term “all personnel” as used within these requirements refers to all personnel deemed to be in scope by the MPoC laboratory.

Assessments to Appendix A are expected to be performed by suitably qualified individuals and cannot be performed as a self-assessment. Remote assessments may be performed to Appendix A, if in line with current PCI SSC guidance on remote assessment processes.

Table 5: Mapping of Appendix A Security Requirements to PCI DSS Requirements

Appendix A Security Requirements	PCI DSS Requirements (from PCI DSS v3.2.1 & PCI DSS v4.0)
A1. Maintain security policies for all personnel	Requirement 12
A2. Secure network connectivity	Requirement 1 Requirement 10 Requirement 11
A3. Develop and maintain secure systems	Requirement 2 Requirement 6
A4. Vulnerability management	Requirement 5 Requirement 6 Requirement 11
A5. Manage access	Requirement 7 Requirement 8
A6. Physical security	Requirement 9
A7. Incident response preparedness	Requirement 10 Requirement 12

A.1 Maintain Security Policies for All Personnel

Security Requirements	Test Requirements	Guidance
Control Objective: Security policies define rules and requirements for all personnel to protect the security and integrity of the entity's resources and protect against identified risks.		
A.1.1 Security governance		
A.1.1.1 Security objectives are aligned with business objectives.	• Examine documentation. • Interview personnel.	The security objectives need to be defined as part of an overarching security strategy that supports and facilitates business objectives. The security strategy needs to provide the foundation for the entity's security policies and procedures and provide a benchmark against which the health of security controls is monitored and measured.
A.1.1.2 Responsibilities and accountability for meeting security objectives are assigned formally, including responsibilities for the security of the environment.	• Examine documentation. • Interview personnel.	The assignment of specific roles and responsibilities need to include monitoring and measurement of performance to ensure security objectives are met. Roles and responsibilities may be assigned to a single owner or multiple owners for different aspects. Ownership needs to be assigned to individuals with the authority to make risk-based decisions and upon whom accountability rests for the specific function. Duties are required to be defined formally, and owners must be able to demonstrate an understanding of their responsibilities and accountability.
A.1.1.3 Responsibility for identifying and addressing evolving risks is assigned.	• Examine documentation. • Interview personnel.	

Security Requirements	Test Requirements	Guidance
A.1.2 Maintain security policies		
A.1.2.1 An organizational security policy(s) is established and disseminated to all relevant personnel.	• Examine documented policies and procedures. • Interview personnel.	A strong security policy, or policies, set the security tone for the entity as a whole and inform personnel what is expected of them. All personnel must be aware of the sensitivity of data and their responsibilities for protecting it. The security policy needs to be updated as needed in response to changes in the environment, results of risk assessments, implementation of new technologies, and changes in business objectives. Personnel must be aware of all policies and policy updates, including their applicable responsibilities. Methods of communicating policies need to include a mechanism for personnel to acknowledge they have received and read the policy or policy update. Personnel acknowledgment may be in writing or electronic. All security policies and policy updates must be approved by management to ensure alignment with the entity's security strategy and business objectives. Any exception to the policies requires management sign-off to ensure the appropriate due diligence is done and approval obtained.
A.1.2.2 Security policies are reviewed and updated as needed to reflect changes to business objectives or the risk environment.		
A.1.2.3 Policy updates are communicated to applicable personnel.		
A.1.2.4 The security policy is approved by management.		
A.1.2.5 An organizational security policy(s) is established and disseminated to all relevant personnel.	• Examine documented policies and procedures. • Interview personnel.	

Security Requirements	Test Requirements	Guidance
A.1.3 Evaluate risk		
A.1.3.1 A risk-assessment process is documented. **A.1.3.2** The documented risk-assessment process is performed at least annually and upon significant changes.	• Examine documented policies and procedures. • Interview personnel responsible for risk-assessment process.	Risks to environments must be assessed at least annually and upon significant changes. The risk assessment is required to identify assets, threats, likelihood, and potential impacts. Risk considerations need to include internal and external attacks—e.g., for cybercrime, fraud, or theft—internal control failures, and malware. Risks must be prioritized, and resources allocated to implement controls that reduce the likelihood and/or the potential impact of the threat being realized. Considerations are required to include regulatory obligations and changes in technology—e.g., deprecation of encryption algorithms. ***Note:*** *Examples of risk-assessment methodologies include but are not limited to OCTAVE, ISO 27005 and NIST SP 800-30.*
A.1.4 Manage risk		
A.1.4.1 A formal risk-management strategy is defined.	• Examine documentation. • Interview personnel.	The risk-management strategy defines a structured approach for identifying, evaluating, managing, and monitoring risk. The strategy is required to include requirements for regularly reviewing and updating the entity's risk-assessment processes as well as methods to monitor the effectiveness of risk-mitigation controls.
A.1.4.2 The risk-management strategy is approved by authorized personnel and updated as needed to address changing risk environment.	• Examine documentation. • Interview personnel.	The risk-management strategy needs to be approved by personnel with appropriate responsibility and accountability.
A.1.5 Manage third-party relationships		
A.1.5.1 Policies and procedures for managing third-party relationships are maintained and implemented.	• Examine documented policies/procedures. • Interview personnel.	Policies and procedures for managing third-party relationships need to consider the risk that each relationship represents, as well as how third-party performance and behavior will be monitored. The policy needs to be kept up to date, approved by management, and communicated to applicable personnel.

Security Requirements	Test Requirements	Guidance
A.1.5.2 Due diligence is performed prior to any engagement with a third party.	• Examine documented procedures. • Examine results of due diligence efforts • Interview personnel.	Due-diligence processes need to include thorough vetting and a risk analysis prior to establishing a formal relationship with the third party. Specific due-diligence processes and goals will vary for each entity, and each is required to provide sufficient assurance that the third party can meet the entity's security and operational needs.
A.1.5.3 Security responsibilities are defined clearly for each third-party engagement.	• Examine documentation • Interview personnel.	The specific approach for defining security responsibilities will depend on the type of service, as well as the particular agreement between the entity and any third parties. The entity needs to have a clear understanding of the security responsibilities to be met by the third party and those to be met by the entity.
A.1.5.4 The entity periodically verifies that the agreed-upon responsibilities are being met.	• Examine results of periodic verification. • Interview personnel.	The specific type of evidence provided by the third party will depend on the agreement in place between the two parties. The evidence needs to provide assurance that the agreed-upon responsibilities are being met on a continual basis. The frequency of verification needs to be aligned with the entity's risk analysis of the service being provided. PCI SSC provides from time to time guidance on best practice methods for management of third parties. The most recent guidance, including for providers of cloud-based services and guidance provided in the PCI DSS requirements, should be referenced when considering this requirement.
A.1.5.5 Written agreements are maintained.	• Examine documentation. • Interview personnel.	Agreements need to promote a consistent level of understanding between parties about their applicable responsibilities and be acknowledged by each party. The acknowledgement evidences each party's commitment to maintaining proper security in regard to the services.

Security Requirements	Test Requirements	Guidance
A.1.6 Educate personnel		
A.1.6.1 A security-awareness program is implemented that provides awareness to all applicable personnel about security policy and procedures.	• Examine documented policies and procedures. • Examine security-awareness materials.	The security-awareness program needs to result in personnel understanding the security policy and procedures, and their responsibilities for following secure processes. All personnel—including full-time, part-time and temporary employees, contractors, and consultants—with access to or the ability to impact the security of the environment must be required to complete training. Training is required upon hire and include periodic refresher sessions at appropriate intervals. The frequency of training needs to be aligned with the entity's policies for education and security awareness, and commensurate with personnel job function.
A.1.6.2 Personnel receive security awareness training at defined intervals, as appropriate for their job function but at least every 12 months.	• Examine records of attendance. • Interview personnel.	
A.1.6.3 Personnel are aware of the security policy and responsibilities as applicable to their job function.	• Interview personnel.	

Security Requirements	Test Requirements	Guidance
A.1.7 Screen personnel		
A.1.7.1 Personnel are screened (background checks) prior to being granted access to the environment. **A.1.7.2** The screening process includes established criteria and a decision process for background check results.	• Examine documented policies and procedures. • Interview personnel. • Examine results of screening process.	The intent of screening personnel is to reduce the risk of fraud and unscrupulous behavior from an internal resource. Role descriptions need to describe the level of security or access required for the role, and the level of screening needs to be appropriate for the particular position. Positions requiring greater responsibility or that have administrative access to critical data or systems may warrant more detailed background checks than positions with less responsibility and access. The policy is required to also cover internal transfers, where personnel in lower risk positions and who have not already undergone a detailed background check are promoted or transferred to positions of greater responsibility or access. The specific roles to be screened depend on the entity's personnel and security policies. For example, an entity may have a policy that requires detailed screening for all personnel or defines different levels of screening for different job functions. Examples of criteria that may be appropriate include employment history, criminal records, credit history, and reference checks.

Security Requirements	Test Requirements	Guidance
A.1.8 Business as Usual (BAU)		
A.1.8.1 Review and/or monitoring is performed periodically to confirm personnel are following security policies and procedures.	• Examine evidence of reviews and/or ongoing monitoring. • Interview personnel.	Periodic reviews and/or ongoing monitoring of personnel and activities is required to ensure that security is included as part of normal business operations on an ongoing basis. Reviews must be performed by responsible personnel as defined by the entity. The frequency of reviews is required to be defined in accordance with the entity's risk assessments and be appropriate for the particular job function.
A.1.8.2 Processes to detect and respond to security control failures are defined and implemented.	• Examine documented processes. • Observe implemented processes. • Interview personnel.	The entity needs to be able to detect any failures in security controls and respond to them in a timely manner. Processes for responding to security control failures is required to include: • Restoring the security control • Identifying the cause of failure • Identifying and addressing any security issues that arose during the failure of the security control • Implementing mitigation, such as process or technical controls, to prevent the cause of the failure recurring • Resuming monitoring of the security control

A.2 Secure Network Connectivity

Security Requirements	Test Requirements	Guidance
Control Objective: Network-connectivity controls provide secure pathways to the entity's systems while protecting those systems from unauthorized access and network-based threats.		
A.2.1 Protect systems from untrusted systems and networks		
A.2.1.1 A security policy(s) and procedures for protection of the environment boundaries are maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	Policies for protecting the environment boundaries need to define the purpose, scope, roles, and responsibilities of defined boundaries to protect system components from untrusted networks. The policy is required to be kept up to date, approved by management, and communicated to applicable personnel.
A.2.1.2 Up-to-date network and data-flow information is maintained for all communication paths.	• Examine network and data-flow information. • Observe methods used to maintain up-to-date network and data-flow information.	Network and data-flow information (e.g., diagrams or network-mapping tools) accurately document how the entity's networks are configured, the identity and location of all systems, how systems are connected to each other, and all communication paths with trusted and untrusted networks.
A.2.1.3 Access between trusted and untrusted networks, systems, and applications is limited via physical and/or logical controls.	• Examine documentation describing controls. • Observe physical and/or logical controls.	Documentation illustrating authorized communications, both internal and external—including source and destination systems, interface connections, security controls for those connections, and the type of data being sent—will assist in meeting these requirements. Protection mechanisms may include technologies such as network gateways, routers, firewalls, encryption, API controls, and virtualization techniques. Controls may be a combination of software and hardware—e.g., use of packet-filtering capability based on header information, advanced filtering/inspection tools, and implementation of dedicated physical network devices or channels to separate network segments. Many security devices and software provide rule sets and settings that can validate the existence and methodologies used to secure network connectivity.
A.2.1.4 Traffic to and from systems is restricted to only that which is necessary, with all other traffic specifically denied.	• Examine documentation identifying necessary traffic. • Observe configurations of ingress and egress controls.	

Security Requirements	Test Requirements	Guidance
A.2.1.5 Network connectivity controls are monitored and/or periodically reviewed to confirm configurations are effective.	• Examine documented methods for monitoring and/or periodically reviewing network connectivity controls. • Observe implemented methods and processes.	Reviewing device configurations allows the entity to identify and remove any unneeded, outdated, or incorrect rules and confirm that only authorized connections, ports, protocols, services, and APIs are allowed as defined in the documented business justifications. All other services, protocols, and ports need to remain disabled or be removed through periodic reviews. Review processes may include real-time monitoring and analysis, periodic maintenance cycles to ensure the controls are accurate and working as intended, and periodic reviews of network traffic connectivity across ports, protocols, and services. For guidance on services, protocols, or ports considered to be insecure, refer to industry standards and guidance—e.g., NIST, ENISA, OWASP, etc.
A.2.2 Protect systems from network threats		
A.2.2.1 Controls are implemented to detect and/or block known and unknown network attacks.	• Examine documented controls/configuration standards. • Observe implemented controls.	Controls must be implemented at the perimeter and critical systems points and include consideration of both network-based and application-based attack vectors. Methods of detection may include signature-based, behavioral, and other mechanisms that analyze traffic flows. Examples of tools include IDS/IPS, host firewalls, and real-time traffic analysis tools. All mechanisms—such as detection engines, baselines, and signatures—must be configured, maintained, and updated per vendor instructions to ensure optimal protection.
A.2.2.2 Suspicious traffic is blocked or generates an alert that is investigated and responded to.	• Examine documented procedures. • Observe implemented controls and processes.	If suspicious traffic is not automatically blocked, an alert should be generated that is actively monitored and immediately investigated. When suspicious traffic is automatically blocked, a record of the traffic needs to be generated and investigated to determine whether action is needed to prevent further attack.

A.3 Develop and Maintain Secure Systems

Security Requirements	Test Requirements	Guidance
Control Objective: Security risks and events can occur at any time during the lifetime of a system or application. Integrating secure processes throughout the lifecycle provides assurance that system integrity is maintained at all times.		
A.3.1 Secure application development		
A.3.1.1 A security policy(s) and procedures for secure management of the Software Development Lifecycle (SDLC) is maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	When software is developed by the entity or bespoke or custom software is developed by a third party for the entity, the software-development process is required to employ secure coding practices to address common vulnerabilities applicable to the particular technology. The entity needs to remain up to date with vulnerability trends and update its secure coding practices and developer training as needed to address new threats. Examples of current best practices include OWASP, SANS CWE Top 25, and CERT Secure Coding. Application developers must be trained properly to identify and resolve issues related to common coding vulnerabilities. Having staff knowledgeable about secure software-development practices minimizes the number of security vulnerabilities accidentally introduced through poor coding practices. Training for developers may be provided in-house or by third parties and needs to be appropriate for the technology used. Common methods for software security testing include threat modeling, code reviews, fuzz testing, and penetration testing. Software security testing should be performed by someone other than the developer of the code to allow for an independent, objective review. Automated tools or processes may also be used in lieu of manual reviews, but keep in mind that it may be difficult or even impossible for an automated tool to identify some coding errors or other security issues. Correcting identified software defects before the software is deployed prevents it from exposing the environments to potential exploit. Requiring a formal review and sign-off by management verifies that the software is approved and has been developed in accordance with policies and procedures.
A.3.1.2 Personnel involved in software development are trained in secure software-development practices.	• Examine evidence of training. • Interview developer personnel.	
A.3.1.3 Software development procedures include processes to address common coding vulnerabilities.	• Examine documented procedures. • Interview developer personnel.	
A.3.1.4 Software security testing is conducted during the software development lifecycle (SDLC) using methodologies documented in the SDLC processes.	• Examine documented software security testing procedures. • Examine results of software security testing. • Interview personnel.	
A.3.1.5 The software security testing process identifies defects and security vulnerabilities.	• Examine documented software security testing procedures. • Examine results of software security testing. • Interview personnel.	
A.3.1.6 Identified software defects and security vulnerabilities are addressed prior to release.		
A.3.1.7 Results of software security testing are signed off by management prior to software release.		

Security Requirements	Test Requirements	Guidance
A.3.2 Configuration standards		
A.3.2.1 A security policy(s) and procedures for system build and configuration management are maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	The policy document needs to explain the purpose, scope, roles and responsibilities, methods of access for different account types, configuration management, monitoring methodology, and controls to address known risks.
A.3.2.2 An up-to-date inventory of all system components in the environment is maintained.	• Examine system inventory. • Interview personnel.	Maintaining a current inventory of all system components enables an organization to accurately and efficiently apply security controls to protect the assets. The inventory should be periodically confirmed by either manual or automated process—e.g., by correlation with the results of vulnerability scans or penetration testing—to confirm it is up to date.
A.3.2.3 Configuration standards are defined and implemented for all system types.	• Examine system configuration standards and build procedures for all system component types. • Examine system configurations. • Interview personnel.	System configuration standards must be kept up to date to ensure that newly identified weaknesses are corrected prior to a system being deployed in the environment. System configuration standards and related processes need to specifically address security settings and parameters that have known security implications for each type of system in use. Examples of industry-accepted configuration standards include, but are not limited to: • Center for Internet Security (CIS) • International Organization for Standardization (ISO) • SysAdmin Audit Network Security (SANS) Institute • National Institute of Standards Technology (NIST) The implemented controls need to provide assurance that all systems in the environment have known secure configurations.
A.3.2.4 Configuration standards address all known security vulnerabilities and are based on industry-accepted system hardening standards.		
A.3.2.5 Configuration standards and build procedures include: • Changing all vendor-supplied default accounts and system settings. • Removing or disabling all unnecessary system or application functionality. • Preventing functions that require different security levels from co-existing on the same system component.	• Examine system configuration standards and build procedures for all system component types. • Examine system configurations. • Interview personnel.	

Security Requirements	Test Requirements	Guidance
A.3.3 Change management		
A.3.3.1 Change-control procedures are defined and implemented for all changes to system components, including "emergency changes."	• Examine documented change-control procedures. • Examine records of changes and compare to system configurations. • Interview personnel.	Defined change-control procedures must be followed for any change that impacts systems in the environment. The impact of the change needs to be documented so that all affected parties can plan appropriately for any processing changes. Changes must be authorized by appropriate parties, as defined by the change-management policy, to verify the change is legitimate. Thorough testing is required to be performed to verify that the security of the environment is not reduced by implementing a change. Back-out procedures need to be documented in case the change fails or adversely affects the security of any system component.
A.3.3.2 All changes are authorized, and the security impact understood prior to implementing the change.		
A.3.3.3 All changes are tested in a non-production environment.		
A.3.3.4 Rollback procedures are prepared for all changes.		
A.3.3.5 Unauthorized changes to system or application configurations are prevented and/or detected and addressed.	• Examine documentation of controls and/or processes. • Observe implemented controls. • Interview personnel.	The implemented process needs to be able to either prevent or detect and address the unauthorized addition, removal, or modification of system-critical files such as configuration file contents, OS programs, and application executables. If the implemented solution relies on detection, processes must be in place to ensure that unauthorized changes are detected and addressed as soon as possible. When unauthorized change attempts are automatically blocked, a record of the attempted change should also be generated and investigated to determine whether action is needed to prevent further attempts. The controls could include change-detection solutions, such as file-integrity monitoring or a frequent re-load of a trusted build to restore the system component to a known secure state.

A.4 Vulnerability Management

Security Requirements	Test Requirements	Guidance
Control Objective: New vulnerabilities are continually being discovered and can enter the network from both internal and external sources. An ongoing cycle of testing and remediation helps ensure that security controls continue to be effective in a changing environment.		
A.4.1 Protect against malicious software		
A.4.1.1 A security policy(s) and procedures for protecting systems against malware are maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	Controls must be implemented to help prevent the introduction and execution of malicious software (malware) on systems in the environment. A combination of methods, tools, and programs may be used—e.g., anti-malware software, application whitelisting, host-based and network-based intrusion-prevention tools, and system instrumentation. A combination of real-time protection and periodic scans should be considered. The implemented controls must be kept current (e.g., updated signatures, baselines, etc.) as applicable for the technology. Anti-malware controls need to remain enabled unless disablement is specifically authorized by management on a case-by-case basis for a limited time period.
A.4.1.2 Controls to prevent and/or detect and remove malicious software are implemented, active, and maintained.	• Examine documented controls/configurations. • Observe implemented controls and processes. • Examine evidence of malware prevention and/or detection and removal. • Interview personnel.	

Security Requirements	Test Requirements	Guidance
A.4.2 Address vulnerabilities and security weaknesses		
A.4.2.1 A security policy(s) and procedures for identifying, ranking, and protecting against vulnerabilities are maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	Policies and procedures define the methods used to identify and remediate vulnerabilities that could affect systems in the environment, and include: • Monitoring vulnerability lists • Performing vulnerability scans and penetration tests • Establishing bug bounty programs Reputable outside sources should be used for security and vulnerability information.
A.4.2.2 Vulnerability scans, both internal and external, are performed at least quarterly to identify and address vulnerabilities.	• Examine vulnerability scanning reports • Interview personnel.	Vulnerability scans and all required remediation are to be completed as frequently as needed to ensure vulnerabilities are addressed in a timely manner. Rescans should be performed to verify vulnerabilities have been addressed. In addition to a regular scanning process, vulnerability scans need to be performed after any significant change to the environment.
A.4.2.3 Vulnerability scans are performed by qualified personnel: • External scans are performed by a PCI SSC Approved Scanning Vendor (ASV). • Internal scans are performed by qualified personnel.	• Examine vulnerability scanning reports. • Interview personnel.	Internal vulnerability scans can be performed by qualified, internal staff or outsourced to a qualified third party. For scans managed by the entity, the entity needs to ensure that scanning engines and vulnerability fingerprints are up to date, and that the scanning engine is configured in accordance with vendor guidance documentation. Internal personnel need to have sufficient knowledge to review and understand the scan results and determine appropriate remediation. Internal personnel who interact with an ASV or other external agency used to perform scanning should also be knowledgeable in the network architecture and implemented security controls to provide the ASV with information needed to complete the scan.
A.4.2.4 Identified vulnerabilities are ranked to determine the criticality of the vulnerability.	• Examine documented procedures for ranking vulnerabilities. • Interview personnel.	Vulnerabilities must be ranked and prioritized in accordance with an industry-accepted methodology or organizational risk-management strategy.

Security Requirements	Test Requirements	Guidance
A.4.2.5 Penetration tests are performed at least annually.	• Examine penetration test reports. • Interview personnel.	Penetration tests must be performed at regular intervals and after significant changes to the environment. The penetration-testing methodology should be based on industry-accepted approaches and incorporate both application-layer and network-layer testing. The scope of testing needs to cover the perimeter and critical systems in the environment and include testing from both inside and outside the network. In addition, testing needs to be performed to verify that all segmentation controls are operational and effective, and that out-of-scope systems and networks do not have access to the environment. The specific methodology, depth, and frequency of the testing should be based on the entity's risk-assessment strategy and be updated as needed to consider new threats and vulnerabilities.
A.4.2.6 Penetration tests are performed by qualified personnel.	• Examine penetration test reports. • Interview personnel.	Penetration testing is to be performed only by qualified personnel who can demonstrate knowledge and experience, and who are organizationally independent of the environment being tested.

Security Requirements	Test Requirements	Guidance
A.4.2.7 Vulnerabilities and penetration testing findings considered as high risk are addressed within one month. All other vulnerabilities and identified security issues are addressed in a timely manner.	• Examine results of penetration tests and vulnerability scanning reports. • Examine evidence of remediation to address vulnerabilities and security issues. • Interview personnel.	Security patches and fixes must be implemented based on risk ranking. When high-risk vulnerabilities cannot be addressed within one month, a formal exception process needs to be followed, including approval by personnel with appropriate responsibility and accountability. After remediation activities have been performed—e.g., implementing a patch or updating a configuration file to address a vulnerability or security flaw—rescans and penetration tests must be performed as necessary to verify the remediation is effective and that the identified vulnerability or security issue has been mitigated. A record of remediation activities should be maintained—e.g., via change-control records, configuration file updates, and audit logs. All updates and patches must be managed in accordance with change-control processes. When applicable, changes to system configurations should be reflected in the configuration build standards.

A.5 Managing Access

Security Requirements	Test Requirements	Guidance
Control Objective: Strong access controls protect systems and data from unauthorized access and can limit the likelihood of a compromised system being used to gain access to other systems and networks.		
A.5.1 Access Management		
A.5.1.1 A security policy(s) and procedures for assigning access are maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	Policies and procedures need to include details of processes for role assignments, oversight processes, business justifications, and user and group privilege controls.

Security Requirements	Test Requirements	Guidance
A.5.1.2 Roles and responsibilities are defined for groups and accounts with access to systems.	• Examine defined roles and responsibilities. • Interview personnel.	Determining who has access to what, for how long, and what level of access they have needs to be based on established roles and responsibilities. This includes the processes used to maintain, monitor, and approve administrative and user access to systems and data.
A.5.1.3 Least privileges are assigned based on individual job function and periodically reviewed.	• Observe assigned access privileges. • Examine evidence that access privileges are periodically reviewed. • Interview personnel.	Access to systems and data needs to be restricted based on business need, while also accounting for the sensitivity of the data being stored, processed, or transmitted. Access privileges must be reviewed by responsible personnel as defined by the entity. The frequency of reviews should be defined in accordance with the entity's defined policies and be appropriate for the level of privilege assigned.
A.5.2 Account management		
A.5.2.1 A security policy(s) and procedures for managing accounts are maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	Established processes and oversight includes approval process for provisioning, monitoring, changing, and revoking accounts with the ability to access a system.
A.5.2.2 Individuals are assigned a unique account ID.	• Examine documented procedures. • Observe account settings.	Assigned unique IDs are required to allow the organization to maintain individual responsibility for actions and an effective audit trail per employee.
A.5.2.3 Controls are implemented to protect the confidentiality and integrity of accounts and credentials.	• Examine documented procedures. • Observe implemented controls.	Implemented controls need to protect the confidentiality and integrity of accounts for both local and remote users. The controls are required to include ensuring that account and credential information is securely transmitted and stored—e.g., using strong cryptography—at all times.
A.5.2.4 Controls are implemented to prevent misuse of accounts.	• Examine documented procedures. • Observe implemented controls.	Processes to prevent misuse of accounts needs to include, at a minimum, the use of account lockouts, lockout durations, session timeouts, and reactivation processes. Inactive user accounts must be removed or disabled within a timely manner. All processes need to align with the entity's security policies and procedures.

Security Requirements	Test Requirements	Guidance
A.5.2.5 Access for third parties is identified, controlled, and monitored.	• Examine documented procedures. • Observe implemented controls.	Configuration and connection requirements must be defined and implemented for all access by third-party personnel. For example, ensuring accounts are enabled only during the time needed and disabled when not in use, and monitoring account activity when in use.
A.5.2.6 Access privileges are monitored and/or reviewed at least quarterly by an authorized individual to confirm access is still required.	• Examine documented processes. • Examine evidence of monitoring and/or reviews. • Interview personnel.	All access privileges must be reviewed regularly, at least quarterly, by an authorized individual. Documentation of reviews needs to be retained. Results of these reviews need to include identification and removal of any unneeded or incorrect access, and to ensure that only individuals with a current business need are granted remote access. Automated processes may be used to assist in reviewing access privileges—e.g., to generate notifications when an account has not been used for a period of time. Organizational processes to actively review and change access when an individual changes job function can also assist.
A.5.3 Authentication		
A.5.3.1 All access to systems requires strong authentication prior to access being granted.	• Examine documented procedures. • Observe implemented controls.	Authentication consists of one or more of: • Something you know, such as a password or passphrase • Something you have, such as a token device or smart card • Something you are, such as a biometric When passwords are used, documented requirements need to include considerations for entropy (strength/complexity), password history and reuse, reset processes, and other best practices for secure password use. Passwords need to meet a minimum level of strength, as defined by the entity's security policy, that provides reasonable assurance they are not guessable and would withstand a brute-force attack.

Security Requirements	Test Requirements	Guidance
A.5.3.2 Multi-factor authentication is required for all access to systems in the environment.	• Examine documented procedures. • Observe implemented controls.	Multi-factor authentication (MFA) requires the completion of at least two different authentication methods (that is, something you know, something you have, and something you are) prior to access being granted. The authentication mechanisms used need to be implemented to ensure their independence such that: • Access to one factor does not grant access to any other factor, and • The compromise of any one factor does not affect the integrity or confidentiality of any other factor. Additionally, no prior knowledge of the success or failure of any factor can be provided to the individual until all factors have been presented. Refer to industry standards and best practices for further guidance on MFA principles. MFA can be applied at the network level, system level, or application level. For example, MFA could be applied when connecting to the secure network or network segment, or when connecting to an individual system component in the environment. MFA is required for all personnel connections to the systems in the environment that occur over a network interface. Examples of access include for purposes of maintenance, configuration, updating, administration, or general management of the systems or networks. MFA is not required for application or system accounts performing automated functions.

A.6 Physical Security

Security Requirements	Test Requirements	Guidance
Control Objective: Individuals with physical access to systems or media could potentially bypass logical access controls and gain access to sensitive assets. Strong physical access controls also protect against the unauthorized addition, modification, removal, or damage of systems and data.		
A.6.1 Restrict physical access		
A.6.1.1 A security policy(s) and procedures for securing physical access to systems is maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	The entity needs to define the physical access controls required to prevent systems from being accessed physically by unauthorized persons. The controls need to cover all physical access points and include procedures for managing onsite employees and third parties. Specific procedures are required for managing visitors, including a visible means for identification and escorts by authorized personnel. Physical access and monitoring controls need to include use of video cameras and/or access-control mechanisms. Data from video cameras and/or access-control mechanisms needs to be logged to provide an audit trail of all physical access to the environment. Access logs must be retained in accordance with the entity's audit log policy. (Refer to Requirement A.7.2.) Monitoring and periodic reviews of physical access controls and audit logs need to be performed to allow early identification of incorrect controls and for timely response to suspicious activities. Personnel are required to be trained to follow procedures at all times. All suspicious activity needs to be managed per incident security procedures. (Refer to Requirement A.7.1.)
A.6.1.2 Facility entry controls are in place to limit and monitor physical access to systems in the environment.	• Observe physical access controls.	
A.6.1.3 Physical access for personnel to the environment is authorized and based on individual job function.	• Examine assigned access permissions. • Interview personnel. • Observe personnel access procedures.	
A.6.1.4 Personnel access is revoked immediately upon termination, and all physical access mechanisms, such as keys, access cards, etc., are returned or disabled.	• Examine documented procedures. • Examine evidence of access revocation and return of physical access mechanisms. • Interview personnel.	

Security Requirements	Test Requirements	Guidance
A.6.2 Secure media		
A.6.2.1 Strict control is maintained over the storage and accessibility of media.	• Observe implemented controls. • Interview personnel.	Controls and processes need to cover secure storage, transport, and disposal of all storage media used in the environment. Procedures and technical controls are needed to provide assurance that media cannot be removed, stolen, or copied by unauthorized persons. The specific controls and level of rigor required to protect media must be appropriate for the sensitivity of the data stored on the media.

A.7 Incident Response Preparedness

Security Requirements	Test Requirements	Guidance
Control Objective: An effective incident-response plan allows an entity to respond to potential security issues quickly and effectively and minimize the potential impact of a security incident or breach.		
A.7.1 Incident-response plan		
A.7.1.1 A security policy(s) and procedures for managing and responding to security incidents is maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	The policy needs to define plans, procedures, and technologies to detect, analyze, and promptly respond to security incidents. Defined procedures need to include response activities, escalation, and notification, and cover all assets and processes that could impact critical operations or data. Procedures must be updated in alignment with operational/business changes and the organization's risk strategy.

Security Requirements	Test Requirements	Guidance
A.7.1.2 An incident-response plan is in place that includes: • Roles and responsibilities • Communication and contact strategies • Specific incident response procedures • Business recovery and continuity procedures • Data back-up processes • Analysis of legal requirements for reporting compromises • Coverage and responses of all critical system components • Consideration of payment brands' response requirements	• Examine documented incident-response plans and procedures. • Interview personnel.	The incident-response plan needs to be comprehensive and include coverage of all systems in the environment. At a minimum, communication and contact strategies need to include notification of the payment brands. Incident response personnel/teams must be trained and knowledgeable in incident-response procedures and be available to respond immediately to an incident.
A.7.1.3 The plan is reviewed and tested at least annually.	• Examine documented procedures. • Examine evidence of reviews and testing. • Interview personnel.	The incident-response plan needs to be reviewed, tested, and updated periodically to incorporate lessons learned. Relevant staff must be included in the testing and be briefed on the post-test review. Testing needs to include validation that system, audit, and monitoring logs are available and contain all needed data.
A.7.2 Audit Logs		
A.7.2.1 A security policy(s) and procedures for generating and managing audit logs is maintained and implemented.	• Examine documented policies and procedures. • Interview personnel.	The policy needs to cover requirements for the generation, collection, management, and retention of audit logs for all system components in the environment.

Security Requirements	Test Requirements	Guidance
A.7.2.2 Audit logs are implemented to: • Link all access to systems to an individual user. • Record security events.	• Examine system configurations. • Observe access attempts. • Examine audit log files.	Recording all personnel access, both physical and logical, to systems in the environment is required to help the entity identify any misuse of accounts and ensure that each individual is accountable for their actions. The determination of "security event" will vary for each organization and may include consideration for the type of technology, location, and function of the system. Logging of security events needs to include notifications or alerts related to suspicious or anomalous activities—e.g., as defined in Requirements A.2.2.2 and A.3.3.5. The level of detail logged is required to be sufficient to identify who, what, where, when, and how an event occurred in the environment.
A.7.2.3 Time synchronization is implemented on systems to ensure system clocks are synchronized and have the correct and consistent time.	• Examine system configurations.	Designated central time server(s) must be defined to receive time signals from trusted external sources based on International Atomic Time or UTC. Central time server(s) should peer with one another to keep accurate time. Systems receive time only from designated central time server(s). Time-synchronization technology needs to be kept current and time data must be protected from unauthorized modification.
A.7.2.4 Logs and security events are monitored and/or reviewed periodically for all systems to identify anomalies or suspicious activity.	• Examine evidence of reviews of logs and security events. • Interview personnel.	Real-time monitoring and/or periodic reviews need to be in place for all security events, critical system logs, and security system logs—e.g., firewalls, IDS/IPS, authentication servers, e-commerce redirection servers, etc. The frequency of reviews should align with the associated risk. The use of log harvesting, parsing, and alerting tools can help facilitate the process by identifying log events that must be reviewed.

Security Requirements	Test Requirements	Guidance
A.7.2.5 Audit logs are secured so they cannot be altered.	• Examine controls/configurations. • Observe attempts to modify audit logs	Only individuals who have a job-related need should be able to view audit log files. Audit logs should be backed up promptly to a centralized log server or media that is difficult to alter. Physical and logical access controls must be in place to prevent unauthorized modifications to audit logs. File-integrity monitoring or change-detection software can be implemented to ensure that any changes to saved log data generate an alert.
A.7.2.6 Audit and monitoring logs are retained for least one year, with a minimum of three months immediately available for analysis.	• Examine audit log files. • Interview personnel.	Log-retention policies need to include storage and retrieval procedures. If stored in off-line locations, procedures need to include assurance that log data can be retrieved in a timely manner. The logs to be retained include at least those defined in Requirement A.7.2.2.

Appendix B Attack Costing Framework

This appendix provides the framework for cost calculation of attacks on the MPoC Solution and/or its components.

It starts with the description of differences between hardware- and software-based tamper-responsive systems, followed by guidelines that must be considered for an attack cost calculation framework. The considerations include the importance of the attack scalability factor as well as its definition, remediation and pre-remediation solutions, and their role and construction of a full attack path. The full attack path rating is constructed based on the tests conducted during the evaluation as per testing requirements.

The rating process is described by listing the steps that the laboratory performs when rating a full attack based on the information collected by partial attack tests and documentation analysis.

The relevant factors that this costing framework considers include attack time, attacker expertise, scalability, knowledge of A&M back-end systems, equipment required for the attack, and COTS device access. The detailed description of all the factors is followed by a discussion about ratings, with examples that show the importance of different factors.

Differences between Hardware and Software Tampering

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

1. **Physical presence and scalability**. A hardware-based tamper-responsive system generally requires an attacker to be physically present, while a software-based tamper-responsive system could be vulnerable to remote attacks. Additionally, the exploitation phase of attacks on software-based tamper-responsive systems could require much less time, expertise, and tools, increasing the scalability of the attack even if the attack requires physical presence.
2. **Layered security**. Like hardware tamper-response systems, software tamper-response systems are based on layered security; however, the number of layers and relevance of each level are much more prominent for software tamper-response systems. This means that multiple levels of protection must be bypassed before an attacker can extract an asset. Therefore, full, and partial attacks must be considered as well as remediation techniques. Additionally, the detection approach of a tamper state is much more distributed in nature. For example, a tamper state might be detected by the back-end attestation and monitoring system with support of the MPoC Application.
3. **Technology**. There are much wider sets of technologies that contribute to the layered security in the case of software-based tamper-responsive systems, such as software-based protection in REE, TEE solutions partially supported by hardware, and SE solutions.

4. **Use of COTS and attack stages**: The use of numerous supporting technologies in a layered security architecture leads to a full attack necessarily consisting of several stages. The following stages are considered:
   a. **Gaining control of the COTS device (rooting).** Gaining control of the platform either by using an OEM supported method or based on existing exploits is often the first stage of an attack. While it is often the first stage, it could be that the MPoC Application itself has an exploitable vulnerability and that full control of the COTS is not needed for the full attack path. Prevention of remote attacks on platforms includes regularly updating the COTS OS.
   b. **Gaining control of the MPoC Application**. Even for a COTS device under full attacker control, several security mechanisms have to be circumvented to gain control of the MPoC Application. Gaining control of the MPoC Application is usually the second stage of the attack. It might be possible to access some of the assets by controlling only the COTS device (COTS-native NFC data, touch signals); however, there might be additional security mechanisms added by the MPoC Application, in which case control of the MPoC Application is needed.
   c. **Asset compromise**. Asset compromise is the goal of the attack and usually attainable after the second stage of the attack. It could be possible that after getting access to the COTS, the assets of the MPoC Application could be extracted from memory without necessarily gaining full control of the MPoC Application. However, the MPoC Application could protect against these attack paths.

Gaining control of the COTS device or device rooting refers to obtaining the necessary privileged level of control on the COTS device for a particular attack. Currently, rooting or jailbreaking is possible in multiple COTS platforms. Therefore, merchants as well as merchant employees could enable root access to a COTS Platform on which an MPoC Application is running.

Remote exploits to gain root access for the most common COTS device types are available for non-patched versions of those COTS platforms. Remote exploits to gain access to updated versions of platforms would require development of zero-day exploit. The effort required for such activity would require a significant amount of work.

The proactive attestation of the platform is the key aspect of the security protection against the attacks that require rooting. Attestation of the platform that relays on hardware mechanisms within the platform (e.g., TEE, Secure processors) usually provide a significant barrier for an attacker.

For attacks that require root privileges, rooting is considered a first step of a full attack, and the estimated effort for identification and exploitation are accounted for based on public knowledge. If sufficiently resistant anti-root mechanisms are present or rooting of the COTS device is not possible for other reasons, the full attack cannot be executed.

Considerations for Attack Cost Calculations

The following considerations exist for this attack framework:

- Identification and exploitation stages
- Scalability rating factor
- Remediation and pre-remediation
- Partial and full attacks

Identification and Exploitation Stages

For an attacker wanting to exploit a vulnerability, the vulnerability must first be identified. This may appear to be a trivial separation, but it is an important one. To illustrate this, first consider a vulnerability that is uncovered following months of analysis by an expert, and a simple attack method published on the Internet. Compare this to a vulnerability that is well known but requires enormous expenditure of time and resources to exploit. Of course, factors such as time need to be treated differently in these cases, and therefore the "cost" required to identify an exploit is included in the costing calculation.

Exploitation involves the actual performance of an attack (partial or full) on a live system.

Scalability Rating Factor

This factor addresses the attack potential that affects large groups of merchants that use the same MPoC Application.

While an attack can be remote, local attacks following a few simple pre-described steps could lead to a scalable attack. Even if physical access is necessary for an effective pre-described attack, the fraud could be performed on multiple merchant COTS devices by an organized group of attackers or merchant employees downloading the attack description from the internet.

Scalability of an attack could depend on the control of the COTS device (usually referred to as "rooting") if that step is required for the full attack. In some cases, rooting can be achieved by the phone owner; however, anti-rooting mechanisms and attestation functions supported by the hardware of different COTS could create a significant obstacle to scaling such an attack.

Control of the COTS device with highest privileges could also be obtained by exploiting the COTS platform remotely. This can potentially lead to scalable attacks for partial attacks that depend on the rooting of the COTS. The only way to prevent such scalable attacks is to have a sufficiently patched system for all approved COTS devices.

As the evaluation laboratory assesses only one version of the MPoC Application, the portability of the attack cannot be assessed on the level of the MPoC Application.

Scalability of the attack is of high importance for MPoC Solutions, as reflected in the number of points given to the scalability factor.

Remediation and Pre-remediation

Remediation provides the MPoC vendor with a solution for attacks against which the MPoC Application cannot defend in the field and renders the solutions sufficiently secure. The remediation step is possible only after successful detection of some portion of an attack, such as changes to the security configuration of the COTS platform, but prior to the exploitation of those changes to expose any assets.

In case of pre-remediation, the product is updated on a regular basis to implement proactive mitigations against known attack paths, or so that the learning curve for the execution of an attack is reset at every update—thereby necessitating a new identification phase for any known attacks. Pre-remediation may resolve the attack attempts that cannot be detected by A&M back-end systems or prevent those attacks that are already known.

Remediation and pre-remediation are not rated, but they determine whether a partial attack is feasible. If an attack requires more time than the pre-remediation cycle takes, or if the attack can be detected and mitigated before the fraud is performed, the partial attack is not considered when constructing the full attack.

Approach to Attack Calculations

Where the MPoC standard requires a costing be provided as evidence for a testing requirement, the costing is to be provided for the specific asset or context of the requirement. For example, a test requirement may require that a cryptographic key be exposed by the attack, and another may require that a PIN be exposed. Each of these attacks are to be considered independently and without consideration for any other assets. An attack that requires that a PIN be exposed is considered successful even if the attack does not also expose the cardholder data associated with that PIN.

When considering an attack, the laboratory may look to chain multiple “partial” attacks—each one used to bypass a specific security control—which can be then chained into a “full” attack that is used to compromise the specific asset covered by the test requirement under assessment.

Partial attacks provide the lab with a possibility to assess strengths of different security measures toward the full attack (i.e., a partial attack is to test-specific security mechanism(s), and a full attack is to compromise a specific assets). It provides the vendor with the analysis of which security measures in the chain are the weakest. Circumventing one countermeasure while having five working together toward a secure product is not a significant increase of risk. Circumventing four out of six reduces the security level significantly and increases the risk toward a full attack.

- **Partial and full attacks**. When the lab assesses a partial attack, the rating factors should be noted to have a clear view for the full attack (combination of multiple partial attacks). After assessing the strength of all security protection features, the laboratory needs to analyze the possibility of a full attack. For a possible full attack, the laboratory will calculate the rating based on the factors determined for all partial attacks as described in the next section.

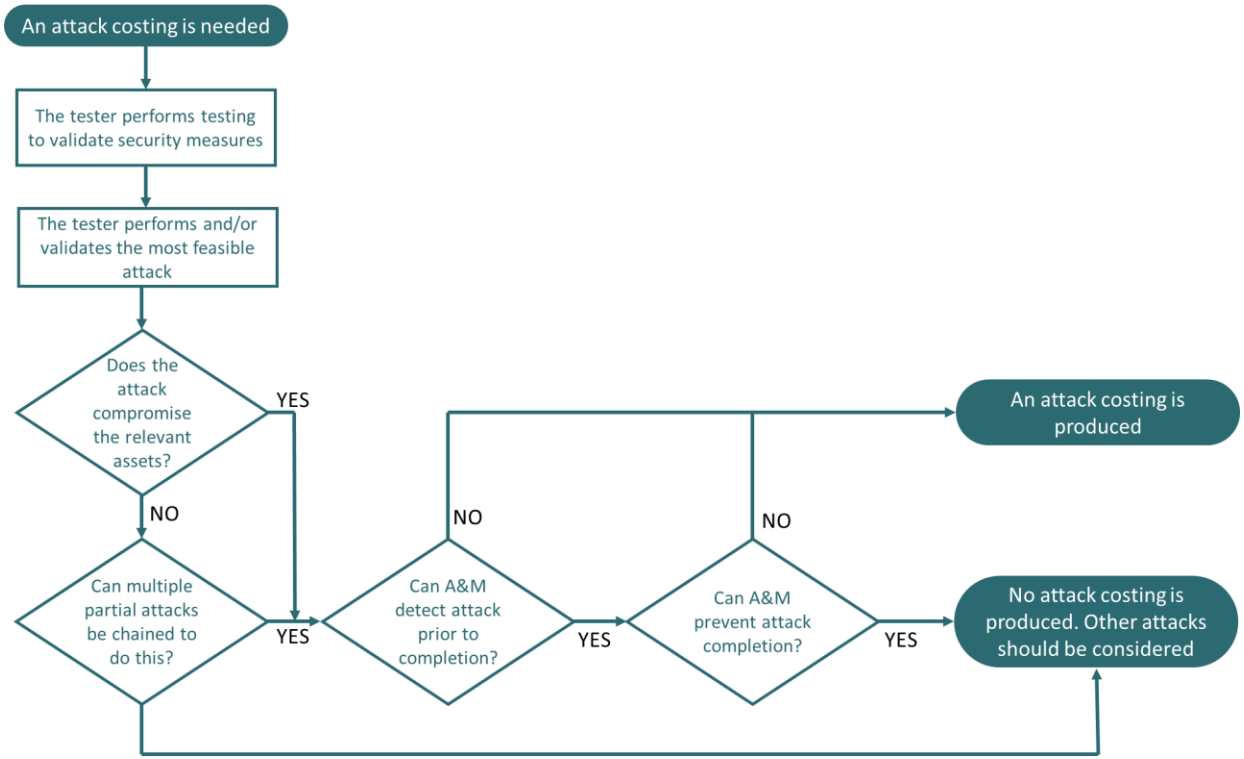
- **Remediation and pre-remediation**. Remediation and pre-remediation can be used to disqualify a partial attack from consideration for rating. The tester, however, has to perform the attack to assess the time needed for the attack development. The attack can be disqualified if the pre-remediation and remediation techniques act faster than the time needed for identification and exploitation. The learning curve from the previous attacks or laboratory expertise must be taken into consideration.

Rating Procedure

The security landscape in which MPoC systems operate dictates a risk-mitigation approach toward the dynamically developing risk landscape. In short, if the security robustness can be circumvented by a skilled attacker de-layering the security architecture, the MPoC Solution could remediate the risk-pre-remediation and remediation techniques discussed in the previous section) and keep the security level sufficiently high.

The rating approach, which considers the attestation and monitoring detection time as well as the remediation time before the costing for a full attack is considered, is shown in Figure 5. The laboratory needs to collect sufficient evidence that attack detection and remediation together are sufficiently strong to prevent an attack, including attacks that use knowledge from any previous attack attempts. Where the laboratory finds that a partial attack is possible, that is some part of the security controls can be bypassed, the laboratory may choose to expose this in their reporting so that the vendor may consider if steps can be taken to improve these security controls.

Figure 5: Attack Costing Rating Procedure for a Full Attack



Where the A&M is not explicitly required to be disabled by the test requirement, consideration is needed with respect to the ability of the A&M to detect and remediate the full attack before it is completed. If testing indicates that the full attack fails because it is detected and successfully remediated prior to the compromise of any assets, then that is not to be considered as a successful attack and no rating is to be given to that attack. In such cases, the laboratory is expected to consider additional or alternative attack paths that would not be detected and remediated.

If the partial attack that circumvents the security protection mechanism cannot be detected and remediated or pre-remediated, the attack can contribute to the full attack path. Therefore, it is necessary to identify the values that different rating factors have so that a full attack rating can be calculated. A full attack comprised of partial attacks without remediation or pre-remediation should be rated. The rating of full attacks based on testing of security components is described below.

If a full attack cannot be constructed, but there are several successful partial attacks, the remaining risk should be identified by the laboratory. Because the security is built from layers of protection measures, the fact that some of the measures can be circumvented reduces the overall security of the solution. Therefore, the remaining risk for the solution should be described.

The relevant factors for attack rating are shown in the following table.

Table 6: Attack Rating Table

Attack Factor	Identification	Exploitation
Attack time		
Attacker Expertise		
Scalability		
Knowledge back-end systems		
Equipment		
COTS device access		
Subtotal		
Total		

As shown in *Figure 5*, partial attacks factors are assessed after the tester performs relevant tests. Partial attacks are analyzed, and the tester determines whether a composition could lead to a full attack. For a possible full attack, the laboratory will calculate the rating based on the factors determined for all partial attacks with the following approach:

- **Attack time**: The time taken for that stage of the attack (identification or exploitation) to be completed.
- **Attacker Expertise**: The partial attack with the highest expertise determines the expertise of the full attack. The full attack would not be possible without all the partial attacks and expertise needed for them.
- **Knowledge back-end systems:** If the knowledge of the back-end systems is needed for any of the partial attacks, the rating of the full attack is determined by the highest level of knowledge needed. The full attack would not be possible without all the partial attacks.
- **Equipment**: The partial attack with the most complex equipment determines the equipment of the full attack. The full attack would not be possible without all the partial attacks and equipment needed for them.

- **COTS device access**: The partial attack with the most difficult category of the COTS device access determines the COTS device access of the full attack. The full attack would not be possible without all the partial attacks and needed COTS device access for each of them.
- **Scalability**: The scalability of the attack is determined based on all the partial attacks that have to be executed. If a step cannot scale easily to a simple and effective attack, the full attack is not scalable.

The categories are rated with values as described in the rest of this section.

Attack Time

The **attack time** is given in the time in hours taken by an attacker to identify or exploit an attack. If the attack consists of several steps, the attack time can be determined and added to achieve a total attack time for each of these steps. Actual labor time must be used instead of time expired as long as there is not a minimum attack time enforced by the attack method applied (for instance, the time needed for performing a side-channel analysis, data collections, or the time needed for an epoxy to harden).

In those cases where attendance is not required during part of the attack time, the attack time is to be taken as expired time divided by 3.

Attack Time	Identification	Exploitation
<1 Day	2	2
<1 week	3	3
<1 month	5	5
>1 month	7	7

Attacker Expertise

The attacker expertise factor determines the needed expertise to perform the attack identification and exploitation. The identified levels of expertise are as follows:

- **Layman**. Persons without professional or specialized knowledge in a particular subject. An attacker who is capable of using an exploit in the form of a script or a written procedure prepared by another attacker. Laymen could be capable of minimal modifications dictated by a simple step process. A layman may be capable of developing an attack if it involves the use of existing platform features (e.g., activating a screen recorder function provided by the OS).
- **Proficient**. Persons who are competent and have the necessary ability, knowledge, and skill to perform customization of attacks successfully. They are familiar with the security functionalities and behavior of the underlying systems. Proficient attackers are capable of flashing OS images and can use root access and management software such as Magisk.

- **Expert**. Persons who are highly knowledgeable and skillful in one or more areas. They are familiar and knowledgeable regarding needed aspects of the solution that could include the underlying OS, algorithms, protocols, hardware components, and physical and logical architectures implemented in the COTS device or system type. This person is capable of developing or introducing complex modifications of an exploit for a specific vulnerability in the system.

Attacker Expertise	Identification	Exploitation
Layman	0	0
Proficient	3	3
Expert	4	4

Scalability

While the time and expertise required to identify and exploit a vulnerability are similar for both hardware and software tamper-responsive systems, it should be noted that a software attack can be re-used much easier on the same or different platforms, even if the local presence is required to initiate the exploitation.

Along with the attack specifics, the following categories are identified:

Scalability	Identification	Exploitation
Instance Specific	N/A	12
Scalable with physical access	N/A	6
Remotely scalable	N/A	0

- **Instance specific:** The attack is not scalable and needs to be heavily customized for each COTS device (e.g., the attack depends on the extraction of keys from each COTS device, and that extraction process must be performed as a bespoke process for each COTS device attacked).
- **Scalable with physical access:** The attack can be leveraged against multiple COTS Platforms and/or MPoC Application instances but requires that the attacker has physical access to the COTS device during the attack. Consideration for if locked or unlocked access is necessary is calculated in the "access" part of the costing.
- **Remotely scalable:** The attack may be leveraged against multiple COTS Platforms and/or MPoC Application instances and does not require the attacker to have any physical access to the COTS device during the attack. Scalability of the attack is of a very high importance for the MPoC; therefore, it has a high number of points.

  Scalability scoring only applies to the exploitation phase of the attack, and a costing that is instance-specific is mostly likely to also require proficient or expert skill during the exploitation stage of the attack (as expertise is required to be used during the attack to enable its application on the instance under attack).

An MPoC assessment must consider the types of COTS Platforms to which the MPoC Application may be deployed, but it is not possible to make determinations during the assessment of how many specific COTS Platforms may be included in any given MPoC Solution. For this reason, the scalability factor is given as two types—scalable (with or without physical access), or instance specific. If an attack is scalable across any given subset of the COTS Platforms supported, it must be considered a scalable attack.

While attacks can be remote, local attacks following a few simple pre-described steps can lead to fraud performed on multiple merchant COTS devices by an organized group of attackers or by merchant employees downloading the description from the internet. Such attacks must also be considered scalable. This type of attack would be costed as "Scalable with physical access."

A scalable attack may also be performed remotely by fooling a merchant into clicking links or installing malicious application and providing the needed privileges to that application. This would be costed as "Remotely scalable."

Scalability of an attack could depend on the control of the COTS device (rooted device) if that step is required for the full attack.

Knowledge of the A&M Back-end Systems

The back-end attestation and monitoring (A&M) is a critical component of the overall software tamper-responsive system. This component is especially critical in a software-based payment, where depending on security and risk management policies, the monitoring system may terminate the payment transaction capability of any MPoC Application instance immediately when there are signs that the COTS device may be compromised. The knowledge of the attestation and monitoring (A&M) can be critical for performing certain attacks. 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 MPoC Application. Identified levels are as follows:

Knowledge of A&M	Identification	Exploitation
Public information	0	0
Restricted information	1	1
Sensitive information	3	4

- **Public information** about the back-end monitoring system (or no information): Information is considered public if it can be easily obtained by anyone (e.g., from the Internet) or if it is provided by the vendor to any customer.
- **Restricted information** concerning the back-end monitoring system (e.g., as gained from vendor technical specifications): Information is considered restricted if it is distributed on request and the distribution is registered.
- **Sensitive information** about the back-end monitoring system—e.g., knowledge of internal design, which may have to be obtained by "social engineering" or exhaustive reverse-engineering.

Equipment

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 two levels are:

- **Standard SW/only** includes pre-existing tools which can be easily obtained, such as Frida, Magisk, or jail-breaking scripts and procedures.
- **Specialized equipment** includes special software that is not readily available and has to be created.

Equipment	Identification	Exploitation
Standard SW/only	0	0
Specialized	3	3

The COTS device security, including the acceptability of any hardware used in the MPoC Solution, is the responsibility of the MPoC Solution provider and their risk management system. Hardware-based security relies on the fact that such solutions have passed high assurance security assessments. The risk management system as well as methods used to determine the acceptability of any hardware security level is assessed by the MPoC laboratory with the documentation assessment based on the relevant security certification as listed in the requirements of this document. Therefore, hardware attacks themselves and hardware equipment used for such attacks are out of scope of this rating system.

COTS Device Access and Remote Attacks

While some attack vectors require only remote access to the COTS device, other attack vectors require local levels of COTS device access. This attack cost factor considers attack vectors that require varying degrees of access to the physical COTS device under attack.

Remote attacks for fully patched platforms are rare and require significant effort and skills. Unpatched systems as well as possible zero-day vulnerabilities could facilitate remote attacks. Zero-day vulnerabilities that provide remote access to the platform are difficult to find for the whole community, let alone for a laboratory during the time of the evaluation. Such attack chains are either very difficult to find and require a lot of time or the expertise to develop attack chains is not readily available.

The unpatched systems have vulnerabilities that are publicly available, and it is the responsibility of the evaluation laboratory to assess the effort of implementing the exploit depending on the type of vulnerabilities as shown in the example ratings within this section. Use or support of unpatched systems can be expected to lead to reduced time, expertise, or increased scalability for an attack—potentially reducing attack ratings to an unacceptable level.

The assessment should consider recently disclosed vulnerabilities, which may lead to potential attacks. It is a requirement of this standard that the A&M systems are constantly updated to monitor for and/or address vulnerabilities that could affect the security of the payment process.

Three types of access are identified:

- Remote – Where no physical access is required.
- Local locked – Where physical access is required, but the device may be in a locked state.
- Local unlocked – Where physical access with the device in an unlocked state is required.

Access	Identification	Exploitation
Remote	0	0
Local locked	0	2
Local unlocked	0	3

Attacks that are marked as “scalable with physical access” will always require local access in the costing as well.

Summary Attack Rating Table

The table below summarizes the factors and costings for an MPoC attack rating.

Attack Time	Identification	Exploitation
<1 Day	2	2
<1 week	3	3
<1 month	5	5
>1 month	7	7
Attacker Expertise	**Identification**	**Exploitation**
Layman	0	0
Proficient	3	3
Expert	4	4
Scalability	**Identification**	**Exploitation**
Instance Specific	N/A	12
Scalable with physical access	N/A	6
Remotely scalable	N/A	0
Knowledge of A&M	**Identification**	**Exploitation**
Public information	0	0
Restricted information	1	1
Sensitive information	3	4
Equipment	**Identification**	**Exploitation**
Standard SW/only	0	0
Specialized	3	3
Access	**Identification**	**Exploitation**
Remote	0	0
Local locked	0	2
Local unlocked	0	3

Discussion of Ratings with Examples

In this section, we present the theoretical attacks of different types and considerations. The attack costing examples show how the assessment of an MPoC Product may differ based on the strength, scalability, and modality (local/remote) of the attack. These costings are provided as examples only and are not necessarily indicative of all costing possibilities for solutions that meet the high-level criteria provided.

The examples provided include the following types of MPoC Solutions:

- A relatively weak MPoC Solution with a scalable attack and one that prevents a scalable attack.
- A relatively strong MPoC Solution with a scalable attack and non-scalable attack.
- An MPoC Solution where there is the ability to use legitimate functionality provided by the COTS OS to expose account data, in both remote and non-remote attack modes.

Example of a Weak Solution

An example of a weak solution has the following evaluation results:

- Weak attestation functions (bypass possible).
- Lack of obfuscation.
- Successful dynamic binary instrumentation (requiring rooted COTS device), which leads to payment assets disclosure.
- Available remote exploit for several approved platforms as well as possibility for merchant and merchant employee to root the COTS device without being detected by the back-end.

The relevant parameters of this attack if it were scalable are:

Attack Factor	Identification		Exploitation	
Attack time	< 1 week	3	< 1 day	2
Attacker Expertise	Proficient	3	Layman	0
Scalability	N/A	0	Scalable	0
Knowledge back-end systems	Public information	0	Public information	0
Equipment	Standard	0	Standard	0
COTS device access	Local unlocked	0	Remote	0
Sub total		6		2
Total	8			

The relevant parameters of this attack if some aspect prevents scalability are:

Attack Factor	Identification		Exploitation	
Attack time	< 1 week	3	< 1 day	2
Attacker Expertise	Proficient	3	Proficient	3
Scalability	N/A	0	Instance specific	12
Knowledge back-end systems	Public information	0	Public information	0
Equipment	Standard	0	Standard	0
COTS device access	Local unlocked	0	Remote	0
Sub total		6		17
Total	23			

Example of a Strong Solution: Scalable and non-Scalable Examples

An example of a strong solution has the following evaluation results:

- It takes an expert 2-3 weeks or more to reverse engineer the solution and circumvent the runtime security mechanisms.
- Obfuscation is very good.
- Rooting is not detected (or rooting detections can be easily circumvented).

The relevant parameters of this attack (for a scalable solution) are:

Attack Factor	Identification		Exploitation	
Attack time	< 1 month	5	< 1 day	2
Attacker Expertise	Expert	4	Layman	0
Scalability	N/A	0	Scalable	0
Knowledge back-end systems	Public information	0	Public information	0
Equipment	Specialized	3	Standard	0
COTS device access	Local unlocked	0	Remote	0
Sub total		12		2
Total	14			

If this attack was instead not scalable (due to MPoC Solution controls, or simply the type of vulnerability exploited) the costing would instead be as below:

Attack Factor	Identification		Exploitation	
Attack time	< 1 month	5	< 1 day	2
Attacker Expertise	Expert	4	Proficient	3
Scalability	N/A	0	Instance specific	12
Knowledge back-end systems	Public information	0	Public information	0
Equipment	Specialized	3	Standard	0
COTS device access	Local unlocked	0	Remote	0
Sub total		12		17
Total	29			

Use and Abuse of Legitimate Functionality Provided by OS

Different attacks exist that use legitimate functionality provided by the OS to gain access to MPoC assets. Examples of such attacks include TeaBot, Cloak, or Dagger using an overlay function as well as accessibility services or developer controls. This example including rating indicates how laboratories should approach such risks during the evaluation.

An example of an abuse of legitimate functionality attack has the following evaluation results:

- Merchant security-guidance documents (user guidance) do not prevent merchant, employees, or attackers from installing a malicious application to monitor touch events, turn on developer options, give the malicious application logging privileges, and run applications in the background.
- The attestation and monitoring mechanism does not provide sufficient information about:
  - The applications that can monitor touch events.
  - Whether the developer options are turned on.
  - Whether malicious applications are running in the background.

- The attack is scalable either in the sense that:
  - Anyone can access and perform these actions on the COTS that runs the application, deeming this attack scalable in the sense of simplified activities that can be performed on multiple sites.
  - The attack can be performed remotely by fooling a merchant into installing malicious application and providing the needed privileges.

The relevant parameters of this attack are:

Attack Factor	Identification		Exploitation	
Attack time	< 1 month	5	< 1 day	2
Attacker Expertise	Proficient	3	Layman	0
Scalability	N/A	0	Scalable	0
Knowledge back-end systems	Public information	0	Public information	0
Equipment	Standard	0	Standard	0
COTS device access	Local unlocked	0	Local unlocked	3
Sub total		8		5
Total	13			

If this attack required the attacker to be physically present to attach a cable and execute a simple set of commands on a locally unlocked device, the costing would instead be as below:

Attack Factor	Identification		Exploitation	
Attack time	< 1 month	5	< 1 day	2
Attacker Expertise	Proficient	3	Layman	0
Scalability	N/A	0	Scalable with physical access	6
Knowledge back-end systems	Public information	0	Public information	0
Equipment	Standard	0	Standard	0
COTS device access	Local unlocked	0	Local unlocked	3
Sub total		8		11
Total	19			

If instead the attack requires more advanced expertise and tooling to expose, and more skill to perform on each device (but still with local unlocked access), the costing would instead be:

Attack Factor	Identification		Exploitation	
Attack time	< 1 month	5	< 1 day	2
Attacker Expertise	Expert	4	Proficient	3
Scalability	N/A	0	Scalable with physical access	6
Knowledge back-end systems	Public information	0	Public information	0
Equipment	Specialized	3	Standard	0
COTS device access	Local unlocked	0	Local unlocked	3
Sub total		12		14
Total	26			

Detailed Examples

This section provides rating examples for two attacks to support the laboratory in rating full attack paths based on its assessment of solution components and construction of the full attack from there. Although rating of a full attack on a platform cannot be tested within the scope of the MPoC evolution, the attack shows the effort needed, as well as the rating that such an attack would receive. Managing risks of attacks on the COTS platforms is handled by the MPoC Solution vendor risk-management system as assessed with this program. The second example is based on testing activities within the scope of this program.

Example of Attack on a Platform: Full Attack – Remote Attack on Platform

***Note**: The attack activities are out of scope for testing activities within MPoC evaluation process.*

The first example describes the attack on the platform that cannot be identified during the MPoC Application assessment by the laboratory. Nevertheless, this attack is included to provide insight into the effort needed for such attacks and show the rating approach. The risk of the remote attacks on the platform should be addressed by the MPoC Solution provider process that manages risks.

The full attack described here could lead to payment assets disclosure by gaining full control of the platform and assuming the MPoC Application is weak and trusts the COTS platform. The attack is based on the real-world attack presented by Samuel Groß (@5aelo), Project Zero in *Full remote attack based on No Clicks Required - Exploiting Memory Corruption Vulnerabilities in Messenger Apps*.

This attack describes what is required to gain remote access and privilege escalation on a platform without known vulnerabilities. The attack description is publicly available.

Attack description: The steps of the attack are:

1. **Reverse engineering**: A researcher starts with the exploration of the MPoC Application structure and protection measures. Sometimes this knowledge is ported from the community. After extracting the contents of the MPoC Application and exploring it, the researcher finds a vulnerability in the NSUnarchiver API that can be triggered without interaction via iMessage, which includes a Shared key dictionary. The vulnerability is registered as CVE-2019-8641 and it is a type of confusion attack.
2. **Understanding platform protection**: The researcher analyzed the solution to identify attack countermeasures. The first countermeasure included separating data and execution parts of the memory and preventing writing in execution parts. However, it would be possible to use existing functions and build return-oriented programming by storing pointers to functions instead. To prevent this attack path, the mitigation techniques of ASLR, Pointer Authentication (PAC), and Code signing have been introduced. The ASLR or address space layout randomization is a countermeasure to prevent return-oriented programming and prevent jumping to predefined addresses. Pointer authentication is present to prevent return-oriented programming. It uses a tag in a pointer to sign and verify pointers.

3. **Building an exploit**: To exploit the vulnerability, the researcher must overcome ASLR as well as PAC. ASLR is overcome by heap spraying with pointers and PAC is handled by creating fake instances of legitimate classes. At this point, the researcher has access to data and camera and can run code outside the sandbox.
4. **Privilege escalation**: The next step is "borrowed" from another researcher (published exploit) and consists of expanding the existing attack with the kernel exploit SockPuppet (CVE-2019-8605 from JavaScript), which has not been patched. This stage of the attack provides access to platform hardware, FS, disabling of code signing, hiding malware, etc.
5. **Conclusion**: At the final stage, the attacker has full control over the platform. For a MPoC Application in the COTS device that suffers such a breach, the payment assets could be available in memory or by developing additional code to tailor the attack to access the COTS-native NFC and send the card data to a remote server.

Comments and rating process. Discovering the parts of this attack, as some parts have been borrowed from other experts, took about 1-to-2 people approximately 3 months. Despite the long identification time, the scalability of the attack indicates it is a lucrative pursuit for attackers. Because the attack is available publicly, if the MPoC Application would support a vulnerable platform, it would be weak against memory dumping and instrumentation, and the A&M would not detect the version and prevent the MPoC Application from working. As a result, the tester would have to rate the attack.

Attack Factor	Identification		Exploitation	
Attack time	> 1 month	7	< 1 day	2
Attacker Expertise	Expert	4	Layman	0
Scalability	N/A	0	Scalable	0
Knowledge back-end systems	Public information	0	Public information	0
Equipment	Specialized	3	Specialized	3
COTS device access	Local unlocked	0	Remote	0
Sub total		14		5
Total	19			

This attack requires the highest level of expertise, consumes significant amounts of time, and results in high scalability on a subset of platforms. This has the potential of affecting significant number of merchants and driving the rating below acceptable levels.

If the A&M system implemented robust controls to prevent this type of attack, it is possible that “sensitive information” would be required in the identification phase (+3 points) as well as proficient skill in the exploitation phase (+3 points) to facilitate bypassing of the A&M controls. Alternatively, new zero-day type attacks may require specialized equipment in terms of the attack software. Therefore, different implementations and attack scenarios may result in similar attacks producing an overall costing of 25 points or above.

Example: Full Attack – Perform Fake Payments

This attack describes what is required to perform faked payments for a specific MPoC Application with several security measures that have been shown to provide limited resistance during laboratory testing.

Attack description. An attacker with MPoC Application control can modify the MPoC Application to simulate payments when the attacker card is detected. The attacker can acquire goods at the attacked merchant and the screen of the COTS device being used will behave normally, displaying the expected information on the screen; however, it will not send the payment to the back-end. A merchant with enough transactions per day will probably identify the theft as shoplifting, as it leaves no trace in the system.

The steps of the attack are:

1. **Rooting the COTS device**. Older Android versions can run the MPoC Application based on the policy of the MPoC Solution. Therefore, the rooting of a COTS device can be performed with standard tooling by unlocking the bootloader and installing a public rootkit. The attestation system does not detect this partial attack.

   The relevant parameters of this partial attack are:

   – **Elapsed time**: <1 day
   – **Attacker Expertise**: Laymen
   – **Scalability**: scalable
   – **Knowledge of** A&M **back-end**: N/A
   – **Equipment**: Standard software
   – **COTS device access**: local locked

2. **Running an MPoC Application on a rooted COTS device**. It is possible to run the MPoC Application on a rooted COTS device as anti-instrumentation. Anti-debugging can be circumvented due to the following weakness found during the evaluation:

   a. **Circumventing runtime protection**. It was identified that runtime protection is limited to some parts of the code, leaving the possibility for the attacker to modify the unprotected code flow relevant for this attack.

   b. **Sensitive information is logged**. The current state of the execution flow of the MPoC Application reveals information such as messages from security components and the current state of the execution flow. This information can aid in attacking the MPoC Application flow.

c. **Monitoring system**. It is possible to circumvent the anti-instrumentation because no information is sent to the back-end about the MPoC Application being compromised. It is possible to perform the transaction after the Frida instrumentation script was ported. The back-end system does not show any reactions caused by the compromise of the MPoC Application or COTS device. User-specific information was not removed from the phone due to the compromise.

   The relevant parameters of this partial attack are:

   - **Elapsed time**: <1 week
   - **Attacker Expertise**: Proficient
   - **Scalability**: scalable
   - **Knowledge of** A&M **back-end**: N/A
   - **Equipment**: Standard software
   - **COTS device access**: local locked

3. **Developing exploit**. Because the obfuscation and run-time integrity are found to be weak during the evaluation, the development of an exploit that modifies the code flow could be performed.

   Circumventing the obfuscation and developing the exploit would require the following parameters:

   - **Elapsed time**: >1 week
   - **Attacker Expertise**: Expert
   - **Scalability**: scalable
   - **Knowledge of** A&M **back-end**: N/A
   - **Equipment**: Standard software
   - **COTS device access**: local locked

   Therefore, the full attack would have the following rating:

   - The elapsed time for identification is accumulated for all the steps resulting in the attack time below a month.
   - The elapsed time for exploitation considers only the activities that must be repeated for other COTS devices with the same platform <1 day.
   - For all other categories, the highest necessary rating is considered because it is the minimum for the full attack to be completed. For example, if one partial attack requires expert while the rest of the partial attacks require only proficient, the full attack cannot be completed without an expert.
   - For exploitation, only the activities that must be repeated are considered in the rating.

   The steps of the attack are costed in a full attack table below:

Attack Factor	Identification		Exploitation	
Attack time	< 1 month	5	< 1 day	2
Attacker Expertise	Expert	4	Layman	0
Scalability	N/A	0	Scalable with physical access	6
Knowledge back-end systems	Public information	0	Public information	0
Equipment	Standard	0	Standard	0
COTS device access	Local unlocked	0	Local locked	2
Sub total		9		10
Total	19			

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

Table 7 lists the minimum key sizes and parameters for the algorithms used with key transport, exchange, or establishment and for data protection in connection with these requirements. Approved key establishment schemes are described in NIST SP800-56A (ECC/FCC3-based key agreement), NIST SP800-56B (IFC-based key agreement) and NIST SP800-38F (AES-based key encryption/wrapping).

Other key sizes and algorithms may be supported for non-payment brand relevant transactions; otherwise, these are the only encryption algorithms designated as Approved Algorithms.

Table 7: Minimum Key Size

Algorithm	IFC (RSA)	ECC (ECDSA, ECDH, ECMQV)	FFC (DSA, DH, MQV)	AES
Minimum key size in number of bits	2048	224	2048/224	128

Equivalent Key Sizes

Key-encipherment keys are to be at least of equal or greater strength than any key they protect, except where explicitly allowed in this standard. This applies to any key-encipherment keys used to protect secret or private keys that are stored, keys used to encrypt any secret, or private keys for loading or transport. For purposes of this requirement, the algorithms and key sizes in each row are considered equivalent. In Table 8:

- RSA key size refers to the size of the modulus.
- Elliptic Curve key size refers to the minimum order of the base point on the elliptic curve. This order is to be slightly smaller than the field size.
- DSA key sizes refer to the size of the modulus and the minimum size of a large subgroup.

Table 8: Equivalent Key Sizes

Algorithm	Effective Bit Strength	IFC (RSA)	ECC (ECDSA, ECDH, ECMQV)	FFC (DSA, DH, MQV)	AES
Minimum key size in number of bits	112	2048	224	2048/224	–
Minimum key size in number of bits	128	3072	256	3072/256	128
Minimum key size in number of bits	192	7680	384	7680/384	192
Minimum key size in number of bits	256	15360	512	15360/512	256

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

- DH implementations entities must generate and distribute the system-wide parameters securely: generator g, prime number p and parameter q, the large prime factor of (p - 1). Parameter p must be at least 2048 bits long and parameter q must be at least 224 bits long. Each entity must generate a private key x and a public key y using the domain parameters (p, q, g).
- 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 at least be as secure as P-224. Each entity must generate a private key d and a public key Q using the specified elliptic curve domain parameters. (See FIPS 186-4 for methods of generating d and Q.)
- Each private key is to be statistically unique, unpredictable, and created using an approved Random Number Generator (RNG), as described in this document. See *Table* 10 for more information.
- Entities are to 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

TLS implementations are used to prevent using cipher suites that do not enforce the use of cryptographic ciphers, hash functions, and key lengths as outlined in this appendix.

KCVs are values used to identify a key without revealing any bits of the actual key itself. Some check values are computed by encrypting an all-zero block using the key or component as the encryption key by using the leftmost n-bits of the result, where n is 40 bits (10 hexadecimal digits or 5 bytes) for AES. Alternatively, AES uses a technique where the KCV is calculated by MACing an all-zero block using the CMAC algorithm as specified in ISO 97971 (see also NIST SP 800-38B). The check value will be the leftmost n-bits of the result, where n is 10 hexadecimal digits. AES is the block cipher used in the CMAC function. The key length of a key or component will be MAC'd using the AES block cipher with an equivalent length (e.g., AES-128 uses 128-bit for MAC while AES-256 uses 256).

Hash Algorithms

For hash algorithms used for authentication or security purposes, only the algorithms and associated bit lengths in Table 9 are permitted.

Table 9: Hash Algorithms

Algorithm	Length
SHA2 family	>255
SHA3 family	>255

RNGs are either a deterministic random number generator or a Non-deterministic Random Number Generator (NRNG).

Random Number Generators

All DRNG must be seeded by an NRNG that provides sufficient authenticated entropy. The entropy required must be at least as many bits as the intended key strength and should be twice as many bits. Entropy sources are discussed in NIST SP800-90B.

Table 10: Random Number Generators

RNG	Requirement
DRNG	Tested and approved under NIST SP 800-90A or ISO/IEC 18031 [§9]
NRNG	Tested and approved under NIST SP 800-90C or ISO/IEC 18031 [§8]

Prime Number Generators

For cryptographic processes that require prime numbers, use prime number generators tested to ISO/IEC 18032 Information Technology--Security Techniques: Prime Number Generation or X9.80 Prime Number Generation, Primality Testing, and Primality Certificates.

Appendix D Secure Software Lifecycle Requirements

This appendix provides a baseline of security requirements with corresponding assessment procedures and guidance to help software vendors that design, develop, and maintain secure software used in the solution throughout the software lifecycle. Vendor's commitment to building capability to develop and maintain secure software is demonstrated by either:

- Evaluation per Appendix D by an approved MPoC Laboratory as part of its evaluation, or
- Through an independent assessment of software lifecycle management practices assessed by a Secure SLC Assessment performed by a Secure SLC Assessor (listed on the PCI SSC website) and confirmed by submitting a Secure SLC ROC and Secure SLC AOC.

Leveraging PCI Secure SLC Standard for Appendix D Security Requirements

The SSF Secure SLC Standard defines security requirements for payment-software vendors to integrate security throughout the entire software lifecycle, resulting in software that is secure by design and better able to withstand attacks. Software vendors whose development processes have been validated as meeting the Secure SLC Standard are listed on the PCI SSC website as a Secure SLC Qualified Vendor.

It is possible to validate compliance to the security requirements in this appendix by confirming that the following are met:

- The software lifecycle management practices were assessed by a Secure SLC Assessor and confirmed to meet all requirements in the Secure SLC Standard with the results documented in a Secure SLC ROC and Secure SLC AOC.
- The software was developed and is being maintained using the software lifecycle management practices that were covered by the Secure SLC assessment.
- A full Secure SLC assessment of the software lifecycle management practices was completed within the previous 36 months. Additionally, if the most recent full Secure SLC assessment occurred more than 12 months ago, an Annual attestation was provided by the developer/vendor within the previous 12 months that confirms continued adherence to Secure SLC Standard for the software lifecycle management practices in use.

D.1 Software Security Governance

A formal software security governance program is established to reflect the vendor's commitment to build secure software, and protect any sensitive assets and resources stored, processed, or transmitted by that software.

Security Requirements	Test Requirements	Guidance
D.1.1 Security Responsibility and Resources Senior leadership team establishes formal responsibility and authority for the security of the vendor's products and services, and resources are allocated to execute the strategy and ensure that personnel are appropriately skilled		
D.1.1.1 Overall responsibility for the security of the vendor's products and services is assigned by the vendor's senior leadership team.	**D.1.1.1.a** Examine vendor evidence and interview the individual or individuals assigned overall responsibility for the security of the vendor's products and services to confirm the following: • Accountability for ensuring the security of the vendor's products and services is formally assigned to an individual or team by the vendor's senior leadership. • Responsibilities include keeping senior leadership informed of security updates, issues, and other matters related to the security of the vendor's products and services. • Updates are provided to senior leadership at least annually on the performance of and changes to the vendor's software security policy and strategy described in Control Objective 2.	The formal assignment of responsibility by the vendor's senior leadership team ensures strategic-level visibility into and influence over the vendor's software security practices. Senior leadership typically represents those individuals or teams with the responsibility and authority to make strategic business decisions for the vendor organization. In many cases, senior leadership teams are comprised of members of the executive team such as the chief executive officer (CEO), chief financial officer (CFO), chief technology officer (CTO), chief information officer (CIO), chief risk officer (CRO), or similar roles, but this is not true for all organizations. Ultimately, the distinct structure of the senior leadership team is determined by the vendor. Assignment of overall responsibility for the vendor's software security program should include the authority to enforce and execute the organization's software-security strategy. Without appropriate authority, those responsible for the security of the vendor's products and services cannot be reasonably held accountable for ensuring that the organization's security strategy is followed. Those responsible for the vendor's software security should provide periodic updates on the state of the vendor's software security program and the performance of its strategy to senior leadership. This allows senior leadership to make sure the strategy is being prioritized and resourced properly, and that changes required as a result of its performance are approved in a timely manner. *(continued on next page)*

Security Requirements	Test Requirements	Guidance
		Evidence to support this control objective might include job descriptions, organization charts, presentations, audio recordings, senior leadership meeting minutes, reports, e-mails, formal communications from senior leadership to the rest of the organization, or any other records that clearly reflect formal assignment of responsibility and authority, and communications between senior leadership and those responsible for the vendor's software security program regarding program performance.
D.1.1.2 Software security responsibilities are assigned.	**D.1.1.2.a** Examine vendor evidence to confirm the following: • Software security responsibilities are clearly defined and assigned to appropriate individuals or teams, including software-development personnel. • Assignment of responsibilities for ensuring the security of the vendor's products and services covers the entire software lifecycle.	Individuals, including third-party personnel, involved in the design, development, testing, and maintenance of the vendor's products and services should be assigned responsibility and accountability for ensuring that software is designed and maintained in accordance with the vendor's security strategy and all applicable security requirements, including software-specific requirements. Responsibilities can be assigned to an individual, group, or role; however, individuals assigned to a particular group or role should clearly understand how those software security responsibilities affect their individual job functions, the organization's security expectations, and the individual's role in fulfilling those expectations. Individuals assigned software-security responsibilities should be able to demonstrate an understanding of their responsibilities and accountability. Evidence to support this security objective might include job descriptions, employee agreements, presentations, company communications, training materials, e-mails, intranet content, or any other documentation or records that clearly and consistently show the assignment of security responsibilities, and the acknowledgement and understanding of those roles and responsibilities.
	D.1.1.2.b Interview a sample of responsible individuals, including software-development personnel, to confirm they are clearly aware of and understand their software security responsibilities.	

Security Requirements	Test Requirements	Guidance
D.1.1.3 Software-development personnel maintain skills in software security matters relevant to their specific role, responsibility, and job function.	**D.1.1.3.a** Examine vendor evidence to confirm the following: • A mature process is implemented and maintained for managing and maintaining software security skills for software-development personnel. • The skills required for each defined role, responsibility, and job function are clearly defined. • The criteria for maintaining individual skills are clearly defined. • The process includes a review at least annually to ensure software-development personnel are maintaining the necessary skills for the security responsibilities they have been assigned.	To be effective in meeting their software security responsibilities, software-development personnel must be trained or have experience in performing such responsibilities and must maintain the appropriate skills to properly carry out those responsibilities. At a minimum, all software-development personnel must have a basic understanding of general software security concepts and best practices. Individuals with specialized roles and responsibilities should additionally possess specialized skills relevant to the functions they perform. Examples of specialized skills include secure software design (software architects), secure coding techniques (software developers), and security-testing techniques (software testers). Efforts to maintain those skills may include vendor-provided training, ongoing participation in local or regional user groups, or the achievement and maintenance of industry-specific certifications. It is up to the vendor to define the necessary criteria for maintaining appropriate job-specific skills and to confirm individual adherence at least annually. Evidence to support this security objective might include policies and processes, training materials or content, records of on-the-job training or course attendance, individual qualification certificates, continuing education credits, or any other documentation or evidence that demonstrates clearly and consistently that software-development personnel possess and maintain appropriate skills and knowledge for their specific job function and responsibilities.
	D.1.1.3.b For a sample of software-development personnel, examine vendor evidence and interview personnel to confirm the following: • Individuals have demonstrated that they possess the skills required for their role, responsibility, or job function. • Individuals have satisfied the criteria for maintaining their individual skills.	
D.1.2 Security Responsibility and Resources Senior leadership team establishes formal responsibility and authority for the security of the vendor's products and services, and resources are allocated to execute the strategy and ensure that personnel are appropriately skilled		

Security Requirements	Test Requirements	Guidance
D.1.2.1 Regulatory and industry security and compliance requirements applicable to the vendor's operations, products, and services and the data stored, processed, or transmitted by the vendor are identified and monitored.	**D.1.2.1** Examine vendor evidence and interview personnel to confirm the following: • A mature process exists to identify and monitor external regulatory and industry security and compliance requirements. • The process includes reviewing sources of regulatory and industry security and compliance requirements for changes at least annually. • The process results in an inventory of external regulatory and industry security and compliance requirements. • The inventory is updated as external security and compliance requirements change.	Many organizations are subject to requirements for protecting certain types of information and data such as personally identifiable information (PII), cardholder data (CHD), and protected health information (PHI). Vendors should maintain awareness of evolving industry and regulatory requirements applicable to their operations and products. Maintaining ongoing awareness of external security and compliance obligations allows the vendor to ensure its processes adequately address those requirements at all times, including whenever those requirements are updated, or new requirements introduced. Evidence to support this control objective might include documented policies and processes, internal standards, requirement mappings, internal presentations, training materials, or any other documentation or records that clearly and consistently show that the vendor has made reasonable efforts to understand and monitor its external security and compliance requirements.
D.1.2.2 A software security policy is defined and establishes the specific rules and goals for ensuring the vendor's products and services are designed, developed, and maintained to be secure, resistant to attack, and in a way that satisfies the vendor's security and compliance obligations.	**D.1.2.2.a** Examine vendor evidence to confirm the following: • A software security policy exists and is communicated to appropriate vendor personnel and business partners, including all software-development personnel. • At a minimum, the policy covers all control objectives within this standard either explicitly or implicitly. • The policy is defined in sufficient detail such that the security rules and goals are measurable. • The vendor's senior leadership team has approved the software security policy. **D.1.2.2.b** Interview a sample of software-development personnel to confirm they are aware of and understand the software security policy.	Vendors must establish a company-wide software security policy to ensure that all individuals or teams - including relevant business partners - involved in software design, development, and maintenance are aware and have a consistent understanding of how the vendor's software products and services should be securely built and maintained, and how any critical assets should be handled. The software security policy (or policies) should be known and thoroughly understood by those with the responsibility to ensure they are met, as well as those individuals and teams who have the ability to affect the security of the vendor's products and services. The vendor's senior leadership team should openly support the establishment and enforcement of the security policy through appropriate communications to vendor personnel, to reinforce the importance of software security to the vendor organization and its leadership.

Security Requirements	Test Requirements	Guidance
D.1.2.3 A formal software-security strategy for ensuring the security of the vendor's products and services and satisfying its software security policy is established and maintained.	**D.1.2.3.a** Examine vendor evidence and interview responsible personnel to confirm the following: • A strategy to ensure the security of the vendor's products and services is defined. • The software-security strategy clearly outlines how the software security policy is to be satisfied. • The software-security strategy is based on or aligned with industry-accepted methodologies. • The software-security strategy covers the entire lifecycle of the vendor's products and services. • The software-security strategy is communicated to appropriate personnel, including software-development personnel. • The software-security strategy is reviewed at least annually and updated as needed, such as when business needs, external drivers, and products and services evolve.	A software-security strategy is a high-level plan, roadmap, or methodology for ensuring the secure design, development, and maintenance of the vendor's products and services, and adherence to the vendor's software-security policy. Vendors should either adopt existing frameworks or methodologies or develop their own in accordance with industry-accepted practices for secure software lifecycle management. By aligning its software-security strategy with industry-accepted methodologies, the vendor is less likely to overlook important aspects of secure software lifecycle management. Vendors that develop their own methodologies should understand how they differ from industry-accepted methodologies, identify any gaps, and ensure that sufficient evidence is maintained to clearly show how their methodologies are at least as effective as those accepted by the industry. Examples of industry-accepted methodologies that are commonly used as benchmarks for secure software development and management include, but are not limited to, current versions of: • *ISO/IEC 27034 Application Security Guidelines* • *Building Security In Maturity Model (BSIMM)* • *OWASP Software Assurance Maturity Model (OpenSAMM)* • *NIST Special Publication 800-160 and its Appendixes* The software-security strategy should evolve as internal factors, such as the vendor's business strategy or product/service offerings or external factors such as external security and compliance requirements, evolve. Therefore, the software-security strategy is not static and should be reviewed and updated periodically to maintain alignment with business needs and priorities. *(continued on next page)*
	D.1.2.3.b Interview a sample of software-development personnel to confirm they are aware of and understand the software-security strategy.	

Security Requirements	Test Requirements	Guidance
		Evidence to support this requirement might include documented security plans or methodologies, presentations, policies and processes, training materials, meeting minutes, interviewer notes, e-mails or executive communications, mappings, or references to industry-accepted methodologies, gap analysis results, or any other records or documentation that clearly and consistently shows that the vendor has made a reasonable effort to develop, maintain, and keep current a formal strategy for satisfying the vendor's software security policy.

Security Requirements	Test Requirements	Guidance
D.1.2.4 Software security-assurance processes are implemented and maintained throughout the entire software lifecycle. ***Note:*** *The focus is on the overall management of security-assurance processes and provides the foundation for specific assurance processes defined within this document.*	**D.1.2.4.a** Examine vendor evidence and interview personnel to confirm the following: • Software security-assurance processes are defined, implemented, and maintained. • An inventory of software security-assurance processes is maintained. **D.1.2.4.b** For a sample of software security-assurance processes, examine vendor evidence and interview personnel to confirm the following: • Software security-assurance processes clearly address the specific rules and goals within the vendor's software security policy. • Software security-assurance processes are aligned with the vendor's software-security strategy. • Vendor personnel, including software-development personnel, are assigned responsibility and accountability for the execution and performance of the security-assurance process in accordance with D.1.1.2. • The individuals or teams responsible for performing and maintaining each security-assurance process are clearly aware of their responsibilities. • The results or outcomes of each security-assurance process are monitored in accordance with D1.2.6.	Software security-assurance processes are activities that are implemented to carry out the vendor's software-security strategy and to facilitate secure software design, development, and maintenance. To ensure that security and compliance requirements are met, software security policy is satisfied, and the vendor's products and services are secure and resistant to attack, vendors need to define such processes throughout all phases of the software lifecycle. These may include security "checkpoints," which are distinct points within the software-development process where software is checked to make sure security requirements are met. Examples of software security-assurance processes and controls include software-design reviews, automated code reviews, security-specific functional testing, and change-management processes. For organizations that leverage Agile software-development methodologies, security checkpoints may be incorporated into the "story" acceptance criteria or the criteria for determining when work is considered "done." Evidence to support this requirement might include documented policies and processes, security-control inventories, output from Governance Risk and Compliance (GRC) or other management tools, software-specific requirements documentation, or any other evidence that clearly and consistently identifies the software security-assurance processes that have been implemented and shows that the security-assurance processes are appropriate for the function they are intended to provide. Additionally, evidence to show the software security-assurance processes are implemented properly may include system or process outputs such as threat models, security test results, bug tracking data, audit log data, incident response, etc.

Security Requirements	Test Requirements	Guidance
D.1.2.5 Evidence is generated and maintained to demonstrate the effectiveness of software security-assurance processes.	**D.1.2.5.a** Examine vendor evidence, including the inventory of software security-assurance processes, and interview personnel to confirm that evidence is generated and maintained for each security-assurance process.	To demonstrate the effectiveness of software security-assurance processes, evidence should be generated and maintained for each process to show that it directly results in or contributes to the expected security outcomes—e.g., fewer vulnerabilities or greater resistance to attacks. Evidence needs to be frequently collected and kept up to date to ensure it accurately reflects the ongoing effectiveness of security-assurance processes. Without a track record of performance for software security-assurance processes, it becomes almost impossible to effectively perform root-cause analysis when such processes fail to produce the expected results. Evidence to support this objective might include security control and evidence generation inventories, vulnerability reports, penetration testing results, or any other records and evidence that clearly and consistently show evidence is generated for each software security-assurance process and that the evidence clearly shows the effectiveness of the processes.
	D.1.2.5.b For a sample of security-assurance processes, examine evidence and other output from the processes and interview personnel to confirm the evidence generated for each process reasonably demonstrates the process is operating effectively and as intended.	

Security Requirements	Test Requirements	Guidance
D.1.2.6 Failures or weaknesses in software security-assurance processes are detected. Weak or ineffective security-assurance processes are updated, augmented, or replaced.	**D.1.2.6.a** Examine vendor evidence and interview personnel to confirm the following: • A mature process exists to detect and evaluate weak or ineffective security-assurance processes. • The criteria for determining a weak or ineffective security-assurance process are defined and justified. • Security assurance processes are updated, augmented, or replaced when deemed weak or ineffective.	Software vendors should monitor their security-assurance processes to confirm the processes remain appropriate—i.e., fit for purpose—and effective for their intended purpose and function. For example, the use of manual code reviews may be sufficient to detect all coding errors and vulnerabilities for software with a very limited code base. However, as the code base grows, the use of manual code reviews for the same purpose becomes increasingly impractical or insufficient, and automated testing tools (such as automated static-code scanners and dynamic software-analysis tools) should be used. One method for detecting weak or ineffective security controls is to define a set of metrics or trends that can be used to measure the effectiveness of security-assurance processes. For example, the results from a vendor's security testing may provide greater insight into the effectiveness of security-assurance processes. If security tests repeatedly find vulnerabilities within the software, it may indicate that applicable security-assurance processes are not being executed properly or working as intended. Another method for detecting weak or ineffective security-assurance processes would be to perform regular reviews of those processes and the evidence generated by those processes to verify they continue to be appropriate for their intended purpose. Evidence to support this requirement might include process-generated evidence, security test results, root-cause analysis, documented remediation actions, or any other evidence that clearly and consistently shows that the effectiveness of software security-assurance processes is monitored, failures and weaknesses are detected, and security-assurance processes are updated, augmented, or replaced when no longer effective at satisfying their intended purpose.
	D.1.2.6.b For a sample of the security-assurance processes identified in D.1.2.4, interview personnel and examine any additional evidence necessary to determine if any failures or weaknesses in those security processes occurred, and to confirm that weak or ineffective processes were updated, augmented, or replaced.	

D.2 Secure Software Engineering

Software is designed and developed to protect critical software assets and to be resistant to attacks.

Security Requirements	Test Requirements	Guidance
D.2.1 Threat Identification and Mitigation Continuously identifies, assesses, and manages risk to its software and services.		
D.2.1.1 Critical assets are identified and classified.	**D.2.1.1.a** Examine vendor evidence to confirm the following: • A mature process exists to identify and classify critical assets. • The criteria for identifying critical assets and determining the confidentiality, integrity, and resiliency requirements for each critical assets are defined. • The process accounts for all types of critical assets—including sensitive assets, sensitive resources, and sensitive functions—for the vendor's software. • The process results in an inventory of critical assets used by the vendor's software.	Before the vendor can determine how to effectively secure and defend its software against attacks, it must first develop a thorough understanding of the software's critical assets that could be targeted by attackers. Critical assets include any sensitive assets collected, stored, processed, or transmitted by the vendor's software, as well as any sensitive functions and sensitive resources within or used by the software. Examples of analysis techniques that could be used to identify critical assets include, but are not limited to, Mission Impact Analysis (MIA), Functional Dependency Network Analysis (FDNA), and Mission Threat Analysis.

Security Requirements	Test Requirements	Guidance
D.2.1.2 Threats to the software and weaknesses within its design are continuously identified and assessed.	**D.2.1.2.a** Examine vendor evidence, including process documentation and assessment results to confirm the following: • A mature process exists to identify, assess, and monitor software threats and design weaknesses (i.e., flaws). • The assessment accounts for all software inputs/outputs, process/data flows, trust boundaries and decision points, and how they may be exploited by an attacker. • The assessment accounts for the entire code base, including how the use of third-party, open-source, or shared components or libraries, APIs, services, and applications used for the delivery and operation of the software may be leveraged in an attack. • The assessment results in a recorded inventory of threats and design flaws. • Assessments are routinely performed to account for changes to existing or the emergence of new threats or design flaws.	Determining how to effectively secure and defend software against attacks requires a thorough understanding of the specific threats and vulnerabilities applicable to the vendor's software. This typically involves understanding: • The motivations an attacker may have for attacking software. • Weaknesses in the software design that an attacker might attempt to exploit. • The exploitability of identified weaknesses. • The impact of a successful attack. This information helps the vendor to identify the threats and design flaws that present the most significant and immediate risk, and to prioritize remediation activities necessary to address them. Information regarding software threats can be obtained from a variety of sources, both external and internal. Examples of external sources include publications from organizations such as SANS, MITRE, and CERT that specialize in tracking common system vulnerabilities and attack techniques, or industry-specific sources that provide threat intelligence for specific sectors, such as FS-ISAC for the financial services industry and R-CISC for the retail industry. Other external sources of threat information and design weaknesses could include technology vendors, open-source user communities, industry publications, and academic papers. Internal sources could include reports from internal research and design teams, formal threat models, or actual activity data from internal security or operations teams. *(continued on next page)*

Security Requirements	Test Requirements	Guidance
	D.2.1.2.b When open-source software components are utilized as part of the software, examine vendor evidence, including process documentation and assessment results to confirm these components are managed as follows: • An inventory of open-source components used in the vendor's software is maintained. • A mature process exists to analyze and mitigate the use of open-source components with known vulnerabilities. • The vendor monitors vulnerabilities in open-source components throughout their use or inclusion in the vendor's software. • An appropriate patching strategy for open-source components is defined.	When open-source software components are used, the vendor should consider any risks associated with the use of the open-source components and the extent to which the open-source software provider manages the security of those components. Additionally, the vendor will need to confirm that support—including up-to-date security patches—is available (whether provided by an internal or external entity) for the open-source component. The use of open-source components should be supported by a clear policy about how those components are evaluated and implemented. A reliable support system should be in place to identify errors or problems and evaluate and address them in a timely manner. When vulnerabilities are identified in open-source components that are applicable to their software, vendors should have processes in place to analyze those vulnerabilities and update the components to appropriate, non-vulnerable versions in a timely manner. When patches for open-source components are no longer available, those components should be replaced by actively supported ones. Vendors should identify and establish sources and processes for managing vulnerabilities in open-source components that are appropriate for their software design and release frequency.
	D.2.1.2.c Examine assessment results for the selected software to confirm the following: • All software inputs/outputs, process/data flows, trust boundaries, and decision points were considered during the assessment. • The entire code base, including how the use of third-party, open-source or shared components or libraries, APIs, services, and applications used for the delivery and operation of the software were considered during the assessment.	

Security Requirements	Test Requirements	Guidance
D.2.1.3 Software security controls are implemented in the software to mitigate threats and design weaknesses.	**D.2.1.3.a** Examine vendor evidence, including process documentation and software-specific threat and design information, to confirm the following: • A mature process exists for defining software-specific security requirements and implementing software security controls within the software to mitigate software threats and design flaws. • Decisions on whether and how to mitigate a specific threat or design flaw are recorded, justified, and approved by appropriate personnel. • Any remaining residual risk is recorded, justified, and approved by appropriate personnel.	To ensure that its software is resistant to attacks, vendors must implement software-specific controls or countermeasures in their software to mitigate the specific threats and design weaknesses. Examples of such controls include the use of multi-factor authentication mechanisms to prevent unauthorized individuals gaining access to critical assets, and logging mechanisms to detect if and when authentication mechanisms might have been circumvented. Other examples include the use of input validation routines or parameterized queries to protect software from SQL-injection attacks. Except where specific software security controls and countermeasures are defined within this standard, it is up to the vendor to determine the most appropriate software security controls to implement. The specific controls used will be dependent on the software threats identified, as well as the software's architecture, the software's intended function, the data it handles, and the external resources it utilizes. Evidence to support this security objective may include software-specific requirements documentation, feature lists, security control inventories, change-management documentation, risk assessment reports, test results, or any other evidence or information that clearly and consistently shows that the vendor implements and maintains security controls in software to address the risks to that software.
	D.2.1.3.b Examine evidence and interview personnel to confirm the following: • Decisions on whether and how to mitigate a specific threat or design flaw are reasonably justified. • Any remaining residual risk is reasonably justified.	
	D.2.1.3.c Examine vendor evidence to confirm that security controls have been implemented to mitigate all identified threats and design flaws.	

Security Requirements	Test Requirements	Guidance
D.2.1.4 Failures or weaknesses in software security controls are detected. Weak or ineffective security controls are updated, augmented, or replaced.	**D.2.1.4.a** Examine vendor evidence and interview personnel to confirm the following: • A mature process exists to identify weak or ineffective software security controls and to update, augment, or replace them. • The criteria for determining a weak or ineffective security control are defined and justified. • The process involves monitoring security control effectiveness throughout the software lifecycle. • Weak or ineffective security controls are updated, augmented, or replaced in a timely manner upon detection.	Vendors should monitor and/or routinely test their software to confirm that implemented software security controls remain appropriate (i.e., fit for purpose) and effective for sufficiently mitigating evolving risks or design flaws. For example, a software-specific security requirement may call for cryptography to be used to protect software communications. While the use of SSL may have been sufficient upon the initial design and release of the software, SSL is no longer sufficient to adequately protect communications as new threats and attack methods have significantly reduced its effectiveness as a security control. Therefore, it is imperative that vendors have processes in place to continuously monitor implemented security controls to make sure that they remain appropriate and sufficient to mitigate evolving threats and design flaws throughout the entire lifetime of the software. Evidence to support this requirement might include software-specific documentation, features lists, software-specific security control inventories, change-management documentation, risk-assessment reports, penetration test results, output from active s, bug bounty program data, or any other evidence or information that clearly and consistently shows that the effectiveness of software security controls is monitored and that software-specific software security controls are updated, augmented, or replaced when no longer effective at satisfying their intended purpose of resisting attacks.
	D.2.1.4.b Examine vendor evidence, including software-specific data or test results, and details of software-specific updates to confirm the following: • Security controls that have been deemed “weak” or “ineffective” have been updated, augmented, or replaced. • Decisions on whether and how to replace and augment weak or ineffective security controls are made in accordance with defined criteria and with D.2.1.3.	

Security Requirements	Test Requirements	Guidance
D.2.2 Vulnerability Detection and Mitigation Detect and mitigate vulnerabilities in software to ensure that its software remains resistant to attacks throughout its entire lifecycle.		
D.2.2.1 Existing and emerging software vulnerabilities are detected in a timely manner.	**D.2.2.1.a** Examine vendor evidence and interview personnel to confirm the following: • A mature process exists for testing software for the existence and emergence of vulnerabilities (i.e., security testing). • Tools or methods used for security testing are appropriate for detecting applicable vulnerabilities in the vendor's software, and are suitable for the software architectures, and the software development languages and frameworks employed. • Security testing is performed throughout the entire software lifecycle, including after release. • Security testing accounts for the entire code base, including detecting vulnerabilities in any third-party, open-source, and shared components and libraries. • Security testing is performed by authorized and objective vendor personnel or third parties. • Security testing results in an inventory of identified vulnerabilities. • Security-testing details including the tools used, their configurations, and the specific tests performed are recorded and retained.	Software should be monitored or routinely tested to confirm that vulnerabilities are identified and mitigated before software or code updates are released into production, and to address any vulnerabilities that may have been discovered since release. Routine security testing should be performed prior to or as part of the code-commit process to detect coding errors or the use of insecure functions. It could also be performed during unit, integration, regression, or interoperability testing, or during separate security testing. Security testing should be performed consistently and throughout all stages of the software lifecycle, including during various pre-release phases of the software-development process and after code release, to ensure the software is free from vulnerabilities upon launch and any subsequent updates, and remains free from vulnerabilities throughout its lifetime. Security testing should be performed by appropriately skilled vendor personnel or third parties. In addition, security testing personnel should be able to conduct tests in an objective way and be authorized to escalate any identified vulnerabilities to appropriate management or development personnel so they can be properly addressed. Evidence to support this security objective could include software-specific requirements documentation, security test results, feature lists, change-management documentation, entries in the vendor's workflow (bug tracking) database, or any other evidence or information that clearly and consistently shows that security testing is performed routinely to detect vulnerabilities in code prior to release as well as vulnerabilities discovered since code launch.

Security Requirements	Test Requirements	Guidance
	D.2.2.1.b Examine evidence, including software-specific security testing configurations and test results to confirm the following: • Security-testing tools are configured in a way that is appropriate for the intended tests performed. • Security testing accounts for the entire code base, including detecting vulnerabilities in any third-party, open-source, and shared components and libraries. • Security testing was performed by authorized and objective vendor personnel or third parties.	
	D.2.2.1.c Examine vendor evidence and interview personnel to confirm that personnel responsible for testing are knowledgeable and skilled in the following areas in accordance with D1.1.3: • Software security testing techniques • Security testing tools settings, configurations, and recommended usage	
	D.2.2.1.d Examine software-specific testing results to confirm that security testing is performed throughout the software lifecycle.	

Security Requirements	Test Requirements	Guidance
D.2.2.2 Newly discovered vulnerabilities are fixed in a timely manner. The reintroduction of similar or previously resolved vulnerabilities is prevented.	**D.2.2.2.a** Examine vendor evidence and interview personnel to confirm the following: • A mature process exists for distributing and deploying fixes for newly discovered vulnerabilities and preventing the reintroduction of previously resolved vulnerabilities. • The process includes methods to prevent previously resolved vulnerabilities or other similar vulnerabilities from being reintroduced into the software. • The criteria for determining the "criticality" or "severity" of vulnerabilities and how to address vulnerabilities are defined and justified. • Fixes to address vulnerabilities in production code are made available and deployed in accordance with defined criteria. • Decisions not to provide fixes in accordance with defined criteria are approved and justified by appropriate personnel on a case-by-case basis.	Vulnerabilities should be addressed in a way commensurate with the risk they pose to the software or its stakeholders. The most critical or severe vulnerabilities (i.e., those with the highest exploitability and/or the greatest impact to stakeholders) should be patched immediately, followed by those with moderate-to-low exploitability and/or impact. Additionally, the discovery of new classes of vulnerabilities should be used as a source of input for process improvement. Software should be reviewed for instances of similar vulnerabilities, and the vendor's development processes updated to enable detection and mitigation of such vulnerabilities in the future. In some cases, it may be impractical for a vendor to fix all identified vulnerabilities prior to the release of production code or updates. In such circumstances, the vendor should have a methodology with clear criteria defined for prioritizing vulnerability fixes. The default outcome should always be that vulnerabilities are fixed before the software is released. In cases where it is not possible to fix a vulnerability prior to release, an exception process involving management at a level commensurate with the severity of the vulnerability should be invoked. The process should include documented justification for why a fix for was not provided to address the vulnerability. If it is not possible to mitigate a certain vulnerability prior to release, the vendor should provide stakeholders with additional guidance to mitigate the risk of exploitation until a security update to fix the vulnerability can be made available. *(continued on next page)*
	D.2.2.2.b For a sample of vendor software, examine software-specific security-testing results and the details of software updates to confirm that security fixes are made available and deployed (where applicable) in **a**ccordance with defined criteria.	
	D.2.2.2.c For the sample of vendor software, interview personnel to confirm that decisions not to provide security fixes in accordance with defined criteria are justified by appropriate personnel.	

Security Requirements	Test Requirements	Guidance
		Under no circumstances should a previously resolved vulnerability be reintroduced into production code, nor should similar vulnerabilities within the same class of vulnerabilities be reintroduced. Additional assurance processes and safeguards should be implemented to ensure that such incidents are avoided. The specific processes to prevent such occurrences will largely depend about how the vendor's software is structured and how the vendor manages software updates. It is up to the vendor to determine the most appropriate methods to prevent the reintroduction of vulnerabilities into production code.

D.3 Secure Software and Data Management

The confidentiality and integrity of software and its critical assets are maintained throughout the software lifecycle.

Security Requirements	Test Requirements	Guidance
D.3.1 Change Management Identify and manage all software changes throughout the software lifecycle.		
D.3.1.1 All changes to software are identified, assessed, and approved.	**D.3.1.1.a** Examine vendor evidence and interview personnel to confirm: • A mature process exists to identify, assess, and approve all changes to software. • The process includes an analysis of the security impact of all changes. • The process results in an inventory of all changes made to software, including a record of the determined security impact. • All change-management decisions are recorded. • All implemented changes are authorized by responsible personnel. • The inventory of changes identifies the individual creator of the code and individual authorizing the change, for each code change. • All decisions to implement changes are justified.	All changes to software should be defined, documented, approved, and tracked so that any vulnerabilities attributed to such changes may be identified and resolved as quickly as possible. The harder it is to trace vulnerabilities back to the changes that introduced them, the longer it takes to resolve those vulnerabilities, which places the software at greater risk of attack or compromise. It is imperative to understand the security risk of a change to the software to ensure that it is addressed accordingly. It often involves understanding the types of software functionality the change impacts (e.g., functionality that deals with encryption or authentication processes), the type of information assets that the functionality can access or manipulate, the likelihood of successful vulnerability exploitation, and the impact a successful attack may have on stakeholders.
	D.3.1.1.b For a sample of changes, examine software-specific and change-specific documentation or evidence to confirm the following: • All changes are authorized by responsible personnel. • All decisions to implement the changes are recorded and include justification for the change. • The inventory of changes clearly identifies the individual creator of the code and the individual authorizing the change, for each code change.	

Security Requirements	Test Requirements	Guidance
D.3.1.2 All software versions are uniquely identified and tracked throughout the software lifecycle.	**D.3.1.2.a** Examine vendor evidence and interview personnel to confirm the following: • A formal system or methodology for uniquely identifying each version of software is defined. • The system or methodology includes arranging unique identifiers or version elements in a sequential and logical way. • All changes to software functionality are clearly associated with a unique software version.	Without a thoroughly defined versioning methodology, changes to software may not be properly identified, and customers and integrators/resellers may not understand the impact of such changes. The system or methodology adopted by the vendor should allow different release versions of a software product to be easily distinguishable. To ensure a software version accurately represents the release version, the versioning system or methodology should be integrated with applicable lifecycle functions, such as code control and change management. The versioning system or methodology should encompass all changes to all relevant software components. As several iterations of a software component may be produced for a single software release, the versioning system or methodology should easily identify the version of each component associated with a specific software release version. The method used for identifying the software release versions—e.g., a version numbering scheme—should be documented and reflect the type of change and its impact on the software.
	D.3.1.2.b For a sample of software updates, examine vendor evidence, including change-specific documentation, to confirm the following: • Software versions are updated in accordance with the defined versioning system or methodology. • All changes to software functionality are clearly associated with a unique software version.	

Security Requirements	Test Requirements	Guidance
D.3.2 Software integrity Protection The integrity of software is protected throughout the software lifecycle.		
D.3.2.1 The integrity of all software code, including third-party components, is maintained throughout the entire software lifecycle.	**D.3.2.1.a** Examine evidence, interview personnel, and observe tools and processes to confirm: • A mature process, mechanism, and/or tool(s) exist to protect the integrity of the software code, including third-party components. • The processes, mechanisms, and/or tools are reasonable and appropriate for protecting the integrity of software code. • Processes, mechanisms, or the use of tools results in the timely detection of any unauthorized attempts to tamper with or access software code. • Unauthorized attempts to tamper with or access software code are investigated in a timely manner.	Effective software-code control practices help ensure that all changes to code are authorized and performed only by those with a legitimate reason to change the code. Examples of these practices include code check-in and check-out procedures with strict access controls, and a comparison—e.g., using a checksum—immediately before updating code to confirm that the last approved version has not been changed. It is important that controls cover all software code, third-party components and libraries, configuration files, etc. that are controlled by the vendor. The integrity and confidentiality of these assets must be maintained, as they often contain sensitive assets such as intellectual property—e.g., business logic—logic of security functions, configuration of cryptographic functions (e.g., software-protected cryptography), etc.
D.3.2.2 Software releases and updates are delivered in a secure way that ensures the integrity of the update code.	**D.3.2.2.a** Examine vendor evidence, interview personnel, and observe tools and processes to confirm the following: • A mature process, mechanism, and/or tool(s) exist to ensure the integrity of software updates during delivery. • The processes, mechanisms, and/or tools are reasonable and appropriate for protecting the update code. • Processes, mechanisms, and/or the use of tools results in the secure delivery of update code.	Effective software-code control practices help ensure that all changes to code are authorized and performed only by those with a legitimate reason to change the code. Examples of these practices include code check-in and check-out procedures with strict access controls, and a comparison—e.g., using a checksum—immediately before updating code to confirm that the last approved version has not been changed. It is important that controls cover all software code, third-party components and libraries, configuration files, etc. that are controlled by the vendor. The integrity and confidentiality of these assets must be maintained, as they often contain sensitive assets such as intellectual property (e.g., business logic), logic of security functions, configuration of cryptographic functions (e.g., software-protected cryptography), etc.

Security Requirements	Test Requirements	Guidance
D.3.3 Sensitive Data Protection The confidentiality of sensitive production data is maintained on vendor systems.		
D.3.3.1 Sensitive production data is only collected and retained on vendor systems where there is a legitimate business or technical need.	**D.3.3.1.a** Examine vendor evidence and interview personnel to confirm the following: • A mature process exists to record and authorize the collection and retention of any sensitive production data. • An inventory of sensitive production data captured or stored by the vendor's products and services is maintained. • Decisions to use sensitive production data are approved by appropriate vendor personnel. • Decisions to use sensitive production data are recorded and reasonably justified.	To protect the confidentiality of any sensitive production data—that is, sensitive assets that are owned by an entity other than the vendor— and stored on vendor systems, such data should never be used for purposes other than those for which the data was originally collected. If the vendor provides services to its customers that could result in the collection of sensitive production data—e.g., for troubleshooting or debugging purposes—the vendor should record those specific data elements it collects and retains, and clearly communicate what data elements are collected and why they are collected to its customers and other relevant stakeholders. The inventory of sensitive production data retained by the vendor should include identification of the specific data elements captured, whether storage of each element is permitted, and the security controls required—e.g., to protect confidentiality and/or integrity—for each data element during storage and transmission.
D.3.3.2 Sensitive production data is protected when retained on vendor systems and securely deleted when no longer needed.	**D.3.3.2.a** Examine vendor evidence and interview personnel to confirm that a mature process exists to ensure sensitive production data is protected when retained on vendor systems and is securely deleted when no longer needed. **D.3.3.2.b** Examine vendor evidence and observe a sample of vendor systems to confirm the following: • Sensitive production data is not resident on vendor systems unless appropriate evidence of approval and justification exists. • Sensitive production data is appropriately protected where it is retained. • Secure deletion processes or mechanisms are sufficient to render sensitive production data irretrievable.	When vendors collect sensitive production data from their customers—e.g., for debugging or other customer support purposes—the vendor should coordinate with its stakeholders to identify which data elements require protection. Vendor stakeholders may have their own definition and associated security requirements for sensitive assets, and appropriate protection efforts should be agreed upon by both parties. When the vendor collects or retains sensitive production data, the vendor should ensure it is secured—e.g., by using robust access control measures and/or strong cryptography with industry-accepted key management processes. As soon as it is no longer needed for its collected purpose, sensitive production data should be securely deleted such that it is not possible to reconstruct or recover the data from any vendor system.

D.4 Security Communications

The vendor provides timely information to its stakeholders (e.g., customers, installers, integrators, etc.) regarding security issues affecting its software, and thorough guidance on secure software implementation, configuration, operation, and updates.

Security Requirements	Test Requirements	Guidance
D.4.1 Vendor Implementation Guidance Stakeholders are provided with clear and thorough guidance on the secure implementation, configuration, and operation of its software.		
D.4.1.1 The vendor provides stakeholders with clear and thorough guidance on the secure implementation, configuration, and operation of its software.	**D.4.1.1.a** Examine vendor evidence and interview personnel to confirm the following: • A mature process exists to produce, maintain, and make available to stakeholders' guidance on the secure implementation, configuration, and operation of its software. • The implementation guidance includes documentation of all configurable security-related options and parameters for the vendor's software, and instructions for properly configuring and securing each of those options and parameters.	When followed, the vendor's implementation guidance provides assurance that the software and patches are securely installed, configured, and maintained in the customer environment, and that all desired security functionality is active and working as intended. The guidance should cover all options and functionality available to software users that could affect the security of the software or the data it interacts with. The guidance should also include secure configuration options for any components provided with or supported by the software, such as external software and underlying platforms. Examples of configurable options include: • Changing default credentials and passwords. • Enabling and disabling application accounts, services, and features. • Changes in resource access permissions. • Integration with third-party cryptographic libraries, RNGs, etc. Following the secure implementation guidance should result in a secure configuration across all configurable options.
	D.4.1.1.b For a sample of vendor software, examine software-specific documentation and materials to confirm that the vendor provides and maintains guidance on the secure configuration of each security-related option or parameter available in the vendor's software.	

Security Requirements	Test Requirements	Guidance
D.4.1.2 Secure implementation guidance includes detailed instructions about how to securely install, configure, and maintain all software components and supported platforms.	**D.4.1.2.a** Examine vendor evidence to confirm the following: • The secure implementation guidance includes instructions about how to securely install or initialize, configure, and maintain the software. • The secure implementation guidance is sufficiently detailed. • Evidence exists or is obtained to show that following the secure implementation guidance results in a secure software configuration.	As the vendor is expected to continuously identify, assess, and manage risks to its software, the vendor's software-change processes should include determining the impact of the change to vendor guidance. Software changes that impact a configurable feature or option should result in an update to the secure implementation guidance.
D.4.1.3 Secure implementation guidance is aligned with software updates.	**D.4.1.3.a** Examine vendor evidence and interview personnel to confirm the following: • The process to produce and maintain secure implementation guidance includes generation of updated guidance when new software updates are released, or security-related options or parameters are introduced or modified. • Secure implementation guidance is reviewed at least annually for accuracy even if updates to security-related options and parameters are not issued.	
	D.4.1.3.b For a sample of software updates, examine secure implementation guidance as well as details of the software updates to confirm that as security-related options and parameters are updated or added, the secure implementation guidance is updated.	

Security Requirements	Test Requirements	Guidance
D.4.2 Stakeholder Communications Communication channels are maintained with stakeholders regarding potential security issues and mitigation options.		
D.4.2.1 Communication channels are defined and made available for customers, installers, integrators, and other relevant parties to report and receive information on security issues and mitigation options.	**D.4.2.1.a** Examine evidence and interview personnel to confirm the following: • A mature process exists to support open, bi-directional communications with stakeholders for reporting and receiving security information regarding the vendor's products and services. • Communication channels provide stakeholders the ability to report security-related issues and to receive timely status updates on their queries. • The vendor maintains resources to respond to reports or inquiries regarding the security of the vendor's products and services.	Vendors should monitor the threat landscape to identify new vulnerabilities and security issues that impact their software on the market. Vendors should also provide open lines of communication to enable researchers or other stakeholders to report newly discovered vulnerabilities in the vendor's products and services. Communication channels could include a publicly disclosed e-mail address, website page, or other method to facilitate interactions with external researchers—e.g., through a formal bug bounty program. The vendor should also maintain teams to respond to such reports and drive processes to fix vulnerabilities in the vendor's software. In addition to supporting the receipt of information about vulnerabilities within its software products, the vendor should also issue communications to customers, installers, and integrators to provide information about known vulnerabilities and when fixes will be available. Fixes/patches should be developed and released in a timely manner, based on criticality and in accordance with D.2.2.2. Vendor security notifications should include the criticality and potential impact of the vulnerability, as well as clear guidance for addressing the vulnerability—e.g., how to install a patch or software update. When a fix is not readily available, the vendor should communicate the risk and provide guidance on mitigation options. *(continued on next page)*
D.4.2.2 Stakeholders are notified about security updates in a timely manner.	**D.4.2.2.a** Examine evidence and interview personnel to confirm a mature process exists to notify stakeholders about security updates in a timely manner.	
D.4.2.3 When security updates are not readily available to address known vulnerabilities or exploits, security notifications are issued to all relevant stakeholders to provide instructions for mitigating the risks associated with the known vulnerabilities and exploits.	**D.4.2.3.a** Examine evidence and interview personnel to confirm that processes include providing stakeholders with instructions for mitigating the threat or reducing the likelihood and/or impact of exploitation of known security issues for which a timely patch is not provided.	
	D.4.2.3.b For a sample of software security updates, examine stakeholder communications, product-specific documentation, security-testing results, and other materials to confirm that where known vulnerabilities are not addressed in the security updates, risk mitigation instructions are provided to stakeholders.	

Security Requirements	Test Requirements	Guidance
		Vendor-initiated communications could include e-mail notifications, website alerts, written notices, social media posts, and any other channels the vendor maintains for stakeholder engagement. Communication channels should be publicized so that stakeholders know how to access them (e.g., by signing up for e-mail notifications). Vendor contact information should also be provided for stakeholders to submit further questions regarding security notifications.
D.4.3 Software Update Information Stakeholders are provided with detailed explanations of all software changes.		
D.4.3.1 Upon release of any software updates, a summary of the specific changes made to the software is provided to stakeholders.	**D.4.3.1.a** Examine evidence and interview personnel to confirm the following: • A mature process exists to communicate all software changes to stakeholders upon software updates. • The process results in a clear and detailed summary of all software changes. • The change summary information clearly outlines the specific software functionality impacted by the changes. • Change details are easily accessible to stakeholders.	Release notes should be provided for all software updates, including details of any impact on software functionality and security controls. Informing stakeholders of the impact of a software update enables them to make informed decisions on whether and when to implement it.
	D.4.3.1.b For a sample of software updates, examine publicly available information or notifications regarding the software updates to confirm the following: • Change summary information is made available to stakeholders. • Change summary information accurately reflects the changes made to the software.	

Appendix E MSR Security Requirements

This appendix provides the security requirements for any non-PTS approved MSR devices used with an MPoC Solution. The security and testing requirements described in this appendix provide a framework for protecting the confidentiality and integrity of account data captured using a non-PTS approved MSR, used in combination with the existing elements of an overall MPoC Solution. Adding optional support to process magnetic-stripe transactions with a magnetic-stripe-only reader allows merchants to use a single solution to accept payments, where magnetic-stripe acceptance is desired but the PCI PTS POI device used with the MPoC Solution does not support MSR functionality.

Only requirements for the non-PTS approved MSR device itself are covered in this Appendix. Requirements for the integration and use of a non-PTS approved MSR device within an overall MPoC Solution are covered within the MPoC requirements listed in the body of this document.

MSR Validation and Testing Requirements

Magnetic-stripe data input to an MPoC Solution must be read by an approved device. This device may be a PCI PTS POI, tested and approved through the PCI PTS POI program, or a non-PTS approved MSR tested according to the requirements in this appendix. Regardless of validation, magnetic-stripe-based transactions may not be used by an MPoC Solution with PIN acceptance.

Many of the testing requirements in this appendix are based on, and reference, the PCI PTS POI v6.x Derived Test Requirements document. However, this appendix references only a subset of the requirements that would be tested against a PCI PTS POI device and exclude all physical security testing requirements. When the PCI PTS POI testing requirements require testing for data or implementations that are not provided by the MSR device—such as PIN acceptance and processing, or physical security—these requirements are not to be assessed.

E.1 Account Data Input

Security Requirements	Test Requirements	Guidance
E.1.1 A non-PTS approved MSR device must only provide for the acceptance of account data through a magnetic readhead interface.	**E.1.1.a** The tester must confirm through examination and observation that the non-PTS approved MSR does not provide any other methods for account data acceptance other than through a magnetic readhead interface.	Support for non-PTS approved MSR devices is provided to accommodate magnetic-stripe cards. Acceptance of any other account data presentment method, such as manual entry, contact or contactless chip, may not be implemented in a non-PTS approved MSR device.

E.2 No Account Data Storage

Security Requirements	Test Requirements	Guidance
E.2.1 A non-PTS approved MSR device must not store account data.	**E.2.1.a** The tester must confirm through examination and observation that the non-PTS approved MSR does not store account data. It must not be possible for more than one set of magnetic-stripe data to be present within the memory of the non-PTS approved MSR device at any one time.	Account data should be transferred from the non-PTS approved MSR device to the MPoC Software as soon as practicable. Although there may be some delay between the read of the data and the transfer of the data to the MPoC Software, there can be only one set of data within the non-PTS approved MSR device at any one time.

E.3 Account Data Processing

Security Requirements	Test Requirements	Guidance
E.3.1 All account data is either encrypted immediately upon entry or entered in cleartext into a secure device and processed within the secure controller of the device.	**E.3.1.a** The tester must evaluate the non-PTS approved MSR to PCI PTS POI v6.x DTR A11 and confirm compliance to all applicable requirements.	Account data should be transferred from the non-PTS approved MSR device to the MPoC Software as soon as practicable. Although there may be some delay between the read of the data and the transfer of the data to the MPoC Software, there can be only one set of data within the non-PTS approved MSR device at any one time.

E.4 Firmware Updates

Security Requirements	Test Requirements	Guidance
E.4.1 Where the device implements firmware, the device must support firmware updates. The device must cryptographically authenticate the firmware and if the authenticity is not confirmed, the firmware update is rejected and deleted. The update mechanism ensures security (i.e., integrity, mutual authentication, and protection against replay) by using an appropriate and declared security protocol when using network connections.	**E.4.1.a** The tester must evaluate the non-PTS approved MSR to PCI PTS POI v6.x DTR B2 and confirm compliance to all applicable requirements.	Support for network connections, or network-based updates, is not required. However, where such updates are supported, then a declared security protocol is required. This security protocol is tested under the requirements of the “Communication and Interfaces” Section (PCI PTS POI v6.x Module 3).

E.5 Protection of Sensitive Services

Security Requirements	Test Requirements	Guidance
E.5.1 Access to sensitive services requires authentication. Sensitive services provide access to the underlying sensitive functions. Sensitive functions are those functions that process sensitive assets such as cryptographic keys, account data, and passwords/authentication codes. Entering or exiting sensitive service shall not reveal or otherwise affect sensitive assets.	**E.5.1.a** The tester must evaluate the non-PTS approved MSR to PCI PTS POI v6.x DTR B5 and confirm compliance to all applicable requirements.	Although PCI PTS POI v6.x B5 includes testing requirements that cover aspects of PIN security, a non-PTS approved MSR device must not allow for the entry or processing of cardholder PINs.

E.6 Sensitive Service Limits

Security Requirements	Test Requirements	Guidance
E.6.1 To minimize the risks from unauthorized use of sensitive services, limits on the number of actions that can be performed, and a time limit shall be imposed, after which the device is forced to return to its normal mode.	**E.6.1.a** The tester must evaluate the non-PTS approved MSR to PCI PTS POI v6.x DTR B6 and confirm compliance to all applicable requirements.	

E.7 Key Management

Security Requirements	Test Requirements	Guidance
E.7.1 The key management techniques implemented in the device conform to ISO11568 and/or ANSI X9.24. When multiple cryptographic keys are supported, key management techniques must support key blocks as defined in DTR B9.	**E.7.1.a** The tester must evaluate the non-PTS approved MSR to PCI PTS POI v6.x DTR B9 and confirm compliance to all applicable requirements.	Implementation of key blocks is not required for devices that support only a single encryption key (i.e., for the encryption of account data from the magnetic-stripe).

E.8 Encryption Mechanism

Security Requirements	Test Requirements	Guidance
E.8.1 All account data shall be encrypted using only ANSI X9 or ISO-approved encryption algorithms (e.g., AES) and should use ANSI X9 or ISO-approved modes of operation.	**E.8.1.a** The tester must evaluate the non-PTS approved MSR to PCI PTS POI v6.x DTR B10 and confirm compliance to all applicable requirements.	Encryption algorithms used to protect account data are required to provide at least 128 bits of effective strength. TDEA encryption is not permitted to be used by a non-PTS approved MSR device for any security services. Use of AES encryption for non-PTS approved MSR devices is recommended.
	E.8.1.b The tester must confirm through examination and observation that the non-PTS approved MSR uses only algorithms and key lengths that provide an effective key strength of at least 128 bits.	

E.9 Remote Access

Security Requirements	Test Requirements	Guidance
E.9.1 If the device can be accessed remotely for the purposes of administration, all access attempts must be cryptographically authenticated. If the authenticity of the access request cannot be confirmed, the access request is denied.	**E.9.1.a** The tester must evaluate the non-PTS approved MSR to PCI PTS POI v6.x DTR B22 and confirm compliance to all applicable requirements.	If the device supports remote access, cryptographic keys used to secure this access are required to be different from the keys used for account data encryption. This is tested under the key management requirements.

E.10 Output of Cleartext Account Data

Security Requirements	Test Requirements	Guidance
E.10.1 There is no mechanism in the device that allows the outputting of cleartext account data.	**E.10.1.a** The tester must evaluate the non-PTS approved MSR to PCI PTS POI v6.x DTR B23 and confirm compliance to all applicable requirements.	Although PCI PTS POI v6.x B23 supports the testing of devices that provide for a non-encrypting mode, non-PTS approved MSR devices are not permitted to implement any mode where account data is output in cleartext.

E.11 Surrogate PAN Values

Security Requirements	Test Requirements	Guidance
E.11.1.a If the device is capable of generating surrogate PAN values to be outputted outside of the device, it is not possible to determine the original PAN knowing only the surrogate value. Where a hash function is used to generate surrogate PAN values, a keyed hash function must be implemented.	**E.11.1.a** Where a hash function is used to generate surrogate PAN values the tester shall verify: a) That an industry standard keyed hash function (such as HMAC) is implemented. b) How the key(s) are loaded and managed for this purpose, and any guidance provided to operators of the device for this purpose.	Support for surrogate PAN values is not required for non-PTS approved MSR devices. Non-PTS approved MSRs tested against this requirement are not required to implement physical tamper detection, and the testing requirements that validate physical security need not be assessed.

E.12 Communications and Interfaces

Security Requirements	Test Requirements	Guidance
E.12.1 The device must provide for secure communications and interfaces, including data origin authentication of encrypted messages and protections against logical anomalies.	**E.12.1.a** The tester must evaluate the non-PTS approved MSR to Module 3 of PCI PTS POI v6.x and confirm compliance to all applicable requirements.	Only applicable Sections of the Module 3 requirements must be assessed. For example, requirements covering the security of IP protocols need not be assessed against devices that do not provide IP connectivity. Regardless of interface types supported, testing of requirements validating the implementation of data origin authentication are always required to be assessed.

E.13 Lifecycle Security Requirements

Security Requirements	Test Requirements	Guidance
E.13.1 The non-PTS approved MSR development, manufacturing, and distribution processes must be compliant to the requirements of PCI PTS POI v6.x Module 4.	**E.13.1.a** The tester must evaluate the non-PTS approved MSR to Module 4 of PCI PTS POI v6.x and confirm compliance to all applicable requirements.	On-site validation of the processes and security controls is not required.

Appendix F 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.

Note: *Prior versions of STS include bugs that have been fixed in the current version. Previous versions must not be used unless the critical fixes present in the current NIST tool have been backported. At a minimum, prior versions must disable the Lempel-Ziv compression test [Hamano 2009] and include fixes to the DFT (Spectral) test [Kim 2004], the Overlapping Template test [Hamano 2007], the Non-Overlapping test [NIST 2014], and the "Proportion of Sequences Passing a Test" test interpretation.*

The tester should request and obtain a sample of 2^{30} bits from the vendor. The tester should exercise care to verify that the vendor-supplied data is interpreted correctly by the STS tool (the STS tool assumes that binary data is in big-endian formatting on all devices).

The STS testing on the data shall be judged as a "pass" if it passes all of the tests, for both the "Proportion of Sequences Passing a Test" interpretation approach and "Uniform Distribution of P-Values" interpretation approach. If the data does not pass all tests, and the failure is marginal, the tester should acquire additional data from the vendor and repeat the testing, including both the initial data and the additional vendor-supplied data.

The STS tool should be configured as per guidance provided in SP 800-22 Revision 1a, which is summarized on the next page.

The following settings are consistent with the SP 800-22 Revision 1a document:

Table 11: Configuration Settings

Configuration Item	Setting	Reference in Key Below
Length of bit streams (n)	1,000,000	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 106 in order to produce meaningful results. The Discrete Fourier Transform (Spectral) test requires n to equal 106. (See SP 800-22r1a Sections 2.8.7, 2.10.7, 2.14.7, 2.15.7, and [NIST 2010].)
Number of bit streams (sample size) (M)	1,073	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 (e.g., further testing to resolve test failures) will necessarily include more bit sequences.
Block Frequency block length	20,000	For the Block Frequency test, if n=106, the test block size should be set between 104 and 106. (See SP 800-22r1a Section 2.2.7.)
Non-Overlapping Templates template length	9	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.
Overlapping Template length	9	The Overlapping test requires selection of a template length of 9 or 10 in order to produce meaningful results. When n=106, the template size of 9 comes closest to fulfilling the parameter selection criteria. (See SP 800-22r1a Section 2.8.7.)
Universal block length (L), number of initialization steps (Q)	L=7, Q=1,280	The Universal test block length (L) and initialization steps (Q) must be consistent with the table in SP800-22r1a Section 2.9.7. For n=106, 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=106, STS automatically uses the parameters recommended here.
Approximate Entropy block length	8	For the Approximate Entropy (ApEn) test, SP 800-22r1a Section 2.12.7 requires the block length to be less than [log2 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.
Serial block length	16	The Serial Test block length is also set based on n. If n=106, the block length must be less than 17. (See SP 800-22r1a Section 2.11.7.)

Configuration Item	Setting	Reference in Key Below
Linear Complexity block length	1,000	The Linear Complexity test block length is required to be set to between 500 and 5,000 (inclusive) and requires that (See SP 800-22r1a Section 2.10.7.)

References

[Rukhin 2010] Rukhin, Andrew, et al., "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," NIST SP 800-22, Revision 1a.

[Kim 2004] Kim, Song-Ju, et al., "Corrections of the NIST Statistical Test Suite for Randomness."

[Hill 2004] Hill, Joshua (InfoGard Labs), "ApEn Test Parameter Selection."

[Hamano 2007] Hamano, K. and Kaneko, T., "Correction of Overlapping Template Matching Test Included in NIST Randomness Test Suite," IEICE Trans. Fundamentals, vol. E90-A, no. 9, pp. 1788-1792, Sept. 2007.

[Hamano 2009] Hamano, Kenji, "Analysis and Application of the T-complexity." Ph.D. thesis, The University of Tokyo.

[NIST 2010] STS Software Revision History. URL: *http://csrc.nist.gov/groups/ST/toolkit/rng/revision_history_software.html*. Internet Archive: *http://web.archive.org/web/20150520193625/http://csrc.nist.gov/groups/ST/toolkit/rng/revision_history_software.html*

[NIST 2014] Current STS Release Notes. URL: *http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html*. Internet Archive: *http://web.archive.org/web/20150103230340/http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html*