Payment Card Industry
3-D Secure (PCI 3DS)

Security Requirements and Assessment Procedures for EMV® 3-D Secure SDK

Version 1.1

December 2018

Table of Contents

Document Changes

Date	Version	Description	Pages
November 2017	1.0	Initial release	
December 2018	1.1	Detailed assessment procedures added	

Introduction

This document, *PCI Security Requirements and Assessment Procedures for EMV® 3-D Secure SDK* (hereafter referred to as the *PCI 3DS SDK Security Standard*), defines security requirements and testing procedures for 3-D Secure (3DS) Software Development Kits (SDK) as defined in the *EMV® 3-D Secure SDK Specification*.

This *PCI 3DS SDK Security Standard* defines security controls to facilitate secure 3DS SDK implementations. It does not address how an entity would meet the requirements in the *EMV® 3-D Secure SDK Specification*. Entities that develop 3DS SDKs (3DS SDK Vendors) may be subject to the requirements in this document. Prior to undergoing an assessment, 3DS SDK Vendors should confirm with the payment brand(s) whether they are required to validate compliance with the security objectives and requirements in this standard.

The requirements in this standard apply specifically to application-based EMV 3-D Secure implementations. Please refer to the *EMV® 3-D Secure Protocol and Core Functions Specification* document for additional information regarding app-based and browser-based EMV 3-D Secure implementations.

Terminology

Definitions for PCI terminology used throughout this document are provided in the general PCI Glossary on the PCI SSC website at https://www.pcisecuritystandards.org/pci_security/glossary. Additionally, Table 1 below includes terms that are used in this PCI 3DS SDK Security Standard:

Table 1: Terms and Definitions

Term	Definition
3-D Secure (3DS)	As defined in the *EMV® 3-D Secure Protocol and Core Functions Specification*: an authentication protocol that enables the secure processing of payment and non-payment card transactions.
3-D Secure (3DS) Requestor App	An application on a consumer device that facilitates a 3-D Secure transaction through the use of a 3DS SDK. The 3DS Requestor App is enabled through integration with the 3DS SDK.
3DS Software Development Kit (3DS SDK)	A component that is incorporated into the 3DS Requestor App. The 3DS SDK performs functions related to 3-D Secure on behalf of the 3DS Server and the Access Control Server.
3DS SDK Laboratory (Lab)	A PCI-recognized laboratory that is qualified by PCI SSC to perform PCI 3DS SDK evaluations.
3DS SDK Vendor	An entity that develops, releases, maintains, and supports 3DS SDKs.

Term	Definition
Access Control Server (ACS)	A component of the 3DS core infrastructure that contains the cardholder authentication rules. The ACS is controlled by the issuer, verifies whether authentication is available for a card number and device type, and authenticates 3DS-enabled transactions.[1]
Attestation of Validation (AOV)	The AOV is a form for 3DS SDK Vendors to attest to the results of a PCI 3DS SDK Security Assessment, as documented in the Report on Validation (ROV).
Deterministic Random Number Generator (DRNG)	An algorithm for generating a sequence of numbers that resembles random numbers but is not considered truly random.
Hook or "hooking" attacks	A technique used to alter the behavior of an operating system, applications, or other software components by intercepting messages or events passed between software components[2].
HTML mode	An HTML interface provided by the 3DS SDK and used to render code to allow a user to interact with 3DS functionality. HTML mode is often associated with the use of the WebView class on Android and the UIViewController class on iOS. Similar functionality may be represented by other terms or classes on other operating systems.
Native mode	An interface native to the underlying operating system and used to render code to allow a user to interact with 3DS functionality.
Reasonable justification or reasonably justified	An explanation of why a decision was made using fair, objective, measurable, and balanced information.
Report on Validation (ROV)	A report documenting detailed results from a PCI 3DS SDK Security Assessment.
Resiliency	The extent to which software can maintain normal operations in adverse conditions, including the ability of software to defend itself from attacks.
Rooted or jailbroken device	A condition where smartphones, tablets, and other devices running mobile operating systems (such as Android or iOS) allow users to obtain privileged control of the operating system's subsystems. "Jailbreaking" is often associated with iOS devices, while "rooting" is typically associated with Android devices. Different terms representing the same concept may be associated with other operating systems, but for the purposes of this standard the terms rooting or jailbreaking are used.
Security testing	Security testing is a process of identifying flaws related to elements of confidentiality, integrity, and resiliency in the assessed system component(s) and security mechanisms. The process usually includes, but is not limited to, activities such as threat modeling, code reviews, vulnerability assessment, penetration testing, fuzz testing, etc.

[1] For further information about 3DS roles and functions, refer to the *EMV® 3-D Secure Protocol and Core Functions Specification*.
[2] Wikipedia contributors. "Hooking." *Wikipedia, The Free Encyclopedia*, 4 Apr. 2017. Web. 8 Jun. 2017

Term	Definition
Sideloading attacks	The act of installing an application obtained from an (untrusted) source other than an official application repository for the device (e.g., the App Store for iOS and Google Play for Android).
Tester	An individual or agent of the PCI 3DS SDK Lab performing the PCI 3DS SDK Assessment (in whole or in part).

The following resources contain additional terminology references:

- *EMV® 3-D Secure Protocol and Core Functions Specification* (www.emvco.com)
- *EMV® 3-D Secure SDK Specification* (www.emvco.com)

Roles and Responsibilities

Several stakeholders are involved in maintaining and managing PCI Standards. The following describes the high-level roles and responsibilities as they relate to the *PCI 3DS SDK Security Standard*:

PCI SSC

PCI SSC maintains various PCI Standards, supporting programs, and related documentation. In relation to the *PCI 3DS SDK Security Standard*, PCI SSC:

- Maintains the *PCI 3DS SDK Security Standard.*
- Maintains supporting documentation, including reporting templates, attestation forms, frequently asked questions (FAQs), and guidance, to assist entities implementing and assessing to the *PCI 3DS SDK Security Standard.*
- Maintains the list of approved 3DS SDK versions and qualified PCI 3DS SDK Labs on the PCI SSC website.
- Maintains a quality assurance program for qualified PCI 3DS SDK 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
- Required validation frequency to a PCI Standard
- Fines or penalties for non-compliance

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.

Scope of Security Requirements

The Security Requirements specified within this document apply to specific 3DS data elements collected by the 3DS SDK in association with 3D-Secure transactions. Not all 3DS data elements associated with 3DS transactions are collected by the 3DS SDK. Table 2 below identifies which specific 3DS data elements are collected by the 3DS SDK and that require protection from unauthorized disclosure (confidentiality) and modification (integrity). Additionally, the 3DS SDK Vendor may also identify other forms of data (for example, application design characteristics, session data, status information, error messages, etc.) that may require protection from unauthorized disclosure or modification. 3DS SDK Vendors should conduct a thorough review of the data stored, processed or transmitted by its 3DS SDK products to identify all forms of data that requires protection, and implement security controls (including those defined within this standard) as appropriate to protect such information.

Table 2: Sensitive 3DS SDK Data Elements

3DS Data Element Type	Description	Protection Requirements	Retention by 3DS SDK Allowed?
Cardholder Data (CHD)	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.	Confidentiality	No
3DS Authentication Data	Includes consumer device information and encrypted device data.	Confidentiality / Integrity	No
3DS (Ephemeral) Public Key Data	Includes the ACS Ephemeral Public Key (Q_T) and the 3DS SDK Ephemeral Public Key (Q_C).	Integrity	No
Internal 3DS Key Material	Internal 3DS SDK ephemeral private keys and session keys.	Confidentiality / Integrity	No
3DS Personal Assurance Data	Information captured by the 3DS SDK during 3DS transactions intended to reflect authenticity of ACS service providers during the challenge flow. Includes issuer logos, certificates, etc.	Confidentiality / Integrity	No
3DS Authentication Challenge Data	Includes information such as the ACS Transaction ID, ACS HTML content, cardholder challenge response data, etc.	Confidentiality / Integrity	No
3DS SDK Reference Data	Information about the 3DS SDK specifically. Includes the 3DS SDK reference number and 3DS SDK Application ID (sdkAppID).	Integrity	Yes
3DS SDK Production Code	Compiled production code for the 3DS SDK	Confidentiality / Integrity	N/A

About This Document

Document Structure

This document is organized into the following two sections:

- **3DS SDK Security Requirements and Assessment Procedures** – The security requirements and assessment procedures that apply specifically to a 3DS SDK version. The following topics are covered under this section:
  - 3DS SDK Integrity Protection
  - Sensitive Information Protection
  - Use of Cryptography
- **3DS SDK Vendor Requirements and Assessment Procedures** – The security requirements and assessment procedures that apply specifically to a 3DS SDK Vendor. The following topics are covered under this section:
  - Risk and Vulnerability Management
  - Stakeholder Guidance

Requirements Architecture

The Security Requirements and Assessment Procedures defined within this standard are presented in the following format:

Security Objective – Identifies the high-level security objective that the 3DS SDK or 3DS SDK Vendor is required to meet. Security objectives are broadly stated to enable 3DS SDK Vendor flexibility in determining the best methods to achieve the stated security objective. However, it is expected that the 3DS SDK Vendor produces clear and unambiguous evidence to illustrate 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 3DS SDK Vendors and PCI 3DS SDK Labs understand the intent behind the security objective.

Requirements – Specific security controls or activities that must be implemented by the 3DS SDK or 3DS SDK Vendor (in addition to any other activities specified by the 3DS SDK Vendor) to support the overarching security objective.

Assessment Procedures – Describe the expected testing activities to be performed by the PCI 3DS SDK Lab to validate whether an 3DS SDK or 3DS SDK Vendor has met a particular security objective and its associated requirements. The assessment procedures are intended to provide the 3DS SDK Vendor and the PCI 3DS SDK Lab with a common understanding of the assessment activities to be performed. The specific methods and items examined, and the personnel interviewed, should be appropriate for the security objective and associated requirements being assessed, and for each 3DS SDK Vendor's particular implementation.

Guidance – Additional information to help 3DS SDK Vendors and PCI 3DS SDK Labs 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.

3DS SDK Assessments and Reporting Procedures

Assessment Process

The focus of the assessment process is the 3DS SDK. It does not require a fully functional 3DS Requester App. The PCI 3DS SDK assessment process typically includes the following steps:

1. The 3DS SDK Vendor submits its 3DS SDK to EMVCo for 3DS SDK Functional Testing, completes the testing and receives a Letter of Approval from EMVCo.
2. The 3DS SDK Vendor contacts the payment brands to determine whether the 3DS SDK is eligible or required to validate compliance with this standard.
3. The 3DS SDK Vendor contracts with a PCI 3DS SDK Lab to perform the PCI 3DS SDK Security Assessment and provides the PCI 3DS SDK Lab its EMVCo Letter of Approval.
4. The PCI 3DS SDK Lab performs the PCI 3DS SDK Security Assessment following the Assessment Procedures for each security objective and associated requirements specified within this standard.
5. The PCI 3DS SDK Lab completes the PCI 3DS SDK Report on Validation (ROV) and Attestation of Validation (AOV) in accordance with applicable PCI templates, guidance, and instructions.
6. The PCI 3DS SDK Lab submits the ROV and AOV, along with any other requested documentation, to both the PCI SSC and applicable payment brand(s).
7. If required, remediation activities are performed by the 3DS SDK Vendor to address security objectives or requirements that are not in place, or security controls that were not sufficiently evidenced. The PCI 3DS SDK Lab will then perform follow-up testing and provide PCI SSC and the applicable payment brand(s) with an updated ROV.
8. Upon receipt of a satisfactory ROV and AOV, PCI SSC will acknowledge the successful completion of the assessment process by recognizing the assessed version of the 3DS SDK on a list of “Approved 3DS SDKs” on the PCI SSC website. Processes for re-assessing updated versions of Approved 3DS SDKs will be documented in an associated PCI Program Guide.

Testing Methods

To facilitate third-party validation of the 3DS SDK, 3DS SDK Vendors must produce clear and sufficient evidence that confirms they have satisfied the security objectives and requirements within this standard. The Assessment Procedures identified for each requirement describe the expected testing activities to be performed to validate whether the 3DS SDK and 3DS SDK Vendor have met the requirements. All Assessment Procedures specified are expected to be performed by a PCI 3DS SDK Lab. Each testing method is described in further detail below:

- **Examine**: The PCI 3DS SDK Lab critically evaluates data evidence. Common examples of such evidence include software design and architecture documents (electronic or physical), source code, configuration and metadata files, and security testing results.
- **Interview**: The PCI 3DS SDK Lab converses with 3DS SDK Vendor personnel. The purpose of interviews may include determining how an activity is performed, whether an activity is performed as defined, and whether personnel have particular knowledge or understanding of specific policies, processes, responsibilities, or concepts.
- **Test**: The PCI 3DS SDK Lab evaluates the 3DS SDK code or the operation of the 3DS SDK using a variety of security testing tools and techniques. Examples of such tools and techniques might include the use of static and dynamic analysis, interactive application security testing, and software composition analysis tools; and techniques such as fuzz testing or penetration testing. It shall be up to the PCI 3DS SDK Lab to determine the specific tools or techniques most appropriate to use to validate whether the 3DS SDK or 3DS SDK Vendor meets a specific 3DS SDK requirement.
- **Observe**: The PCI 3DS SDK Lab observes an activity or views something within the software or execution environment. An example includes observing the software perform a function or respond to input to confirm the 3DS SDK is operating as expected.

The Assessment Procedures provide both the 3DS SDK Vendors and PCI 3DS SDK Labs with a common understanding of the expected assessment activities to be performed. The specific items to be examined, observed, or analyzed, and personnel to be interviewed should be appropriate for the requirement being assessed and for each 3DS SDK Vendor's unique 3DS SDK products. It is at the discretion of the PCI 3DS SDK Labs to determine the appropriateness or adequacy of the evidence provided by the vendor to support each requirement.

When documenting the assessment results, the PCI 3DS SDK Lab identifies the testing activities performed and the results of each activity. While it is expected that the PCI 3DS SDK Lab will perform all Assessment Procedures identified for each requirement, it may also be possible for a requirement to be validated using different or additional assessment procedures. In such cases, the lab should document why assessment procedures that differ from those identified in this standard were used, and how those assessment procedures provide at least the same level of assurance as would have been achieved using the assessment procedures defined within this standard. Where terms such as "periodic," "appropriate," and "reasonable" are used in the assessment procedures, it is the 3DS SDK Vendor's responsibility to define and defend its approach to satisfying applicable requirements. However, it is ultimately up to the PCI 3DS SDK Lab whether to accept the vendor's justification given the risks applicable to the vendors 3DS SDK product and the extent to which the 3DS SDK Vendor has mitigated those risks.

Required Vendor Materials

To support validation that the 3DS SDK and 3DS SDK Vendor meet the security objectives and requirements defined within this standard, the 3DS SDK Vendor must provide sufficient evidence to enable a PCI 3DS SDK Lab to validate the requirements. Such evidence may include, but not be limited to, source code, formal documentation such as policies and procedures, or informal documentation such as design documents, data-flow diagrams, process descriptions, and results of internal analysis or testing as defined by the 3DS SDK Vendor. Any such evidence must clearly and concisely illustrate that the security controls implemented by the 3DS SDK or 3DS SDK Vendor facilitate conformance with the security objectives and requirements. Such evidence must also illustrate the ongoing effectiveness of those security controls. Exceptions should also be documented in accordance with the "Exceptions" section below.

Additionally, the 3DS SDK Vendor must provide full access to (1) all un-obfuscated source code and (2) all obfuscated code for all internally developed functionality as well as bespoke or custom functionality developed by third parties. Failure to provide adequate access to source code shall be considered a failure to meet applicable security objectives and requirements.

Exceptions

In some cases, it may be impossible for a 3DS SDK or 3DS SDK Vendor to meet a specific requirement as stated. In such cases, the 3DS SDK Vendor must provide clear and unambiguous justification for why the requirement cannot be met. In order to be considered compliant to the requirements within this standard, the 3DS SDK Vendor must also provide evidence to clearly illustrate that the corresponding security objective is still being met and that other functionality or methods are employed to provide equal or greater assurance to that provided by the methods described in the requirement. 3DS SDK Vendors should work with their PCI 3DS SDK Labs to determine the evidence required to satisfy a specific security objective or associated requirement.

Supporting Multiple Platforms and Versions

3DS SDKs for different operating systems and major versions represent different 3DS SDKs. Updates to 3DS SDKs to support new or different operating systems or major versions are required to undergo new EMVCo Functional and PCI 3DS SDK security assessments for each new operating system and major version supported. Refer to the *EMV® 3-D Secure SDK Specification* for further information on support for multiple platforms and versions.

Test Harness

To facilitate testing of the 3DS SDK in accordance with the Assessment Procedures defined within this standard, a test harness must be provided to verify the 3DS SDK's compliance to the applicable security objectives and requirements. A test harness is considered to be a special test functionality that is either separate or absent from production-level code. The test harness could be developed by the 3DS SDK Vendor, a PCI 3DS SDK Lab, or by another qualified third party. The 3DS SDK Vendor should work with its PCI 3DS SDK Lab to determine the materials required for the assessment of a particular 3DS SDK.

The test harness must rely on as much underlying intended production-level functionality as possible. The test harness serves the purpose of providing a test framework that allows for the 3DS SDK functionality to be exercised outside a production-level deployment environment (such as within a 3DS Requestor App) to verify the 3DS SDK's compliance to the applicable security objectives and associated requirements. For example, elevated privileges or access capabilities may need to be granted for the purpose of providing run-time visibility into various facets of the 3DS SDK's functionality. Another example is providing a test function to initiate a test 3DS transaction.

Component Integration

If the 3DS SDK leverages security services from components defined within the 3DS SDK architecture that reside outside the formal technical boundary of the 3DS SDK (for example, at the application, OS, or device level), those security services will also require validation. 3DS SDK Vendors that utilize these services are responsible for obtaining the necessary evidence and materials to support validation of these components. For example, if the 3DS SDK utilizes a random-number generating function provided by the OS, the vendor is responsible for obtaining and providing evidence as part of the 3DS SDK evaluation that the function has been evaluated to confirm output is sufficiently random, such as validation against ISO/IEC 24759 or via the listing of the software library or the hardware module as a Validated FIPS 140-1 and FIPS 140-2 Cryptographic Module (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm). Moreover, as part of the 3DS SDK evaluation, the PCI 3DS SDK Lab must evaluate the interaction between the 3DS SDK and the external security services components.

3DS SDK Security Requirements and Assessment Procedures

Security Objective 1: Protect the Integrity of the 3DS SDK

To protect the sensitive information handled by the 3DS SDK and to facilitate secure and trustworthy 3DS SDK transactions, the 3DS SDK must implement measures to defend itself in what must be assumed to be a hostile environment (such as in a mobile application operating on a consumer mobile device). Some of the key risks associated with mobile applications and components include threats associated with a "rooted" or "jailbroken" device and threats from other applications operating within the same environment (and with access to shared resources). Appropriate detective and protective mechanisms must be implemented to ensure that the integrity of the 3DS SDK and sensitive 3DS SDK data elements is maintained. Refer to Table 2, "Sensitive 3DS SDK Data Elements," in the "Scope of Security Requirements" section of this document for more information on which specific 3DS SDK data elements require protection from unauthorized modification.

Requirements	Assessment Procedures	Guidance
Requirement 1: Mechanisms are implemented to prevent unauthorized modification of the 3DS SDK, the functionality it provides, and the sensitive 3DS SDK data elements it handles.		
1.1 Security Checks The 3DS SDK provides functionality to conduct device security checks (at a minimum, during initialization) and makes the results of those checks available to the 3DS Requestor App upon request as warning messages. Device security checks include, at minimum, determining whether: • The device is rooted or jailbroken • An emulator is being used to run the 3DS SDK • The 3DS SDK has been tampered with • A debugger is attached	**T.1.1.1** The tester shall examine vendor materials and other evidence to determine what features are provided by the 3DS SDK to provide security checks of the operating environment, and how any issues are escalated to the 3DS Requestor App. From this review, the tester shall determine what APIs are provided by the 3DS SDK for these purposes, the functions that are performed (and how often), and what error/warning messages are returned by the 3DS SDK. At a minimum, the SDK shall perform the checks upon initialization. **T.1.1.2** Based on the information provided in T.1.1.1, the tester shall examine vendor evidence, including source code, to determine what methods are used to perform the environment validation, and what they are designed to detect. These checks must include the below, at a minimum: • Checking whether the device is rooted or jailbroken • Determining whether an emulator is being used to run the 3DS SDK • Validation of the integrity of the 3DS SDK, to determine whether it has been tampered with • Determining whether a debugger is attached or the device running the 3DS SDK is in a "developer" or "debug" mode	The purpose of device security checks is to collect potential indicators of device compromise and make that information available to the ACS to allow the issuer to make risk-based decisions on how to proceed with the 3DS transaction. The security checks are performed during the initialization of the 3DS SDK and the results are passed to the ACS via the 3DS Requestor App along with other device-specific information. For more information on the interactions between the 3DS SDK, the Requestor App, and the ACS, refer to the EMV® 3-D Secure SDK Specification. Possible indicators of compromise include the existence of specific executables or other residual files and artifacts created during the rooting or jailbreaking process. Reflection can be used on Android and iOS to calculate a checksum on the 3DS SDK.

Requirements	Assessment Procedures	Guidance
	T.1.1.3 The tester shall also determine the platforms, operating systems, and specific versions of these supported by the 3DS SDK.	
	T.1.1.4 The tester shall test the 3DS SDK by attempting to execute the 3DS SDK on phones that have been rooted or jailbroken, and observe the response of the SDK to confirm that the 3DS SDK detects that the phone has been rooted or jailbroken. This test must include use of devices from at least two different manufacturers where the 3DS SDK or operating system supports it.	
	T.1.1.5 The tester shall test the 3DS SDK by attempting to execute the 3DS SDK within an environment where raised privileges have been attained (i.e., the system has been rooted or jailbroken) with at least two readily available rooting or jailbreaking tools to confirm that the 3DS SDK detects such elevated privileges.	
	T.1.1.6 The tester shall test the 3DS SDK by attempting to execute the SDK within at least two different emulators and observe the response of the 3DS SDK to confirm that the 3DS SDK detects when an emulator is being used to run the 3DS SDK.	
	T.1.1.7 The tester shall test the 3DS SDK by attempting to modify the 3DS SDK (for example: by modifying initialization files, runtime files, or cryptographic keys) and then execute this modified version to confirm the 3DS SDK detects when its code or execution has been tampered with. The modification must be specifically designed to attempt to bypass the integrity checking of the 3DS SDK.	
	T.1.1.8 The tester shall test the 3DS SDK by attempting to execute the 3DS SDK in a system that allows for advanced control or insight into the execution of applications—e.g., such as running it in debug or developer mode to confirm the 3DS SDK detects this behavior.	

Requirements	Assessment Procedures	Guidance
1.2 Installed from Approved Source The 3DS SDK conducts checks (at minimum, during initialization) to determine whether the embedding Requestor App was installed from a trusted app store (for example: Google Play, iTunes, and Microsoft App store, etc.), and makes that information available to the Access Control Server (ACS).	**T.1.2.1** The tester shall examine vendor materials and other evidence to confirm features are provided by the 3DS SDK to check that the Requestor App was installed by a trusted app store. This shall include confirming that action is taken by the 3DS SDK if the Requestor App fails this test.	While device and operating system vendors have certain control over the applications installed on the mobile device, the end users always have the option to install (also referred to as "sideloading") applications from non-vetted sources (for example, via USB or SD Card, received as an e-mail attachment or downloaded from a file-sharing site). Applications sideloaded from untrusted resources are often packed with malware and request (from the end user) elevated privileges, allowing it to access sensitive information stored on the device or hijack end-user interaction with legitimate resources. To ensure that the Requestor App was not sideloaded, the 3DS SDK should perform checks (based on the operating system) to determine whether the application was installed from a trusted app store. Checks may include interrogating the device OS for installed application packages (for example, using "PackageManager" class on Android) or validating the app signature at run time.
	T.1.2.2 Based on the information provided in T.1.2.1, the tester shall examine vendor materials and other evidence, including source code, to confirm the 3DS SDK verifies the Requestor App was not sideloaded or otherwise installed outside of the primary application distribution store of the operating system being used.	
	T.1.2.3 The tester shall test the 3DS SDK by attempting to sideload an application and have it interface to the 3DS SDK as required by the 3DS SDK documentation for a valid Requestor Application. The tester shall observe the response of the SDK to confirm that normal 3DS transaction processing is not permitted/performed by the 3DS SDK.	
	T.1.2.4 The tester shall examine publicly available information to determine what methods, if any, may exist to bypass the checks performed. Using this information, the tester shall test the 3DS SDK by attempting to perform such bypass using a sideloaded application to confirm that normal 3DS transaction processing is not permitted/performed by the 3DS SDK.	
1.3 Run-Time Integrity The 3DS SDK performs run-time integrity checks to detect when its functionality has been modified. ***Note:*** *These checks shall go beyond the integrity checks performed during initialization as part of Requirement 1.1, "Security Checks."*	**T.1.3.1** The tester shall examine vendor materials and other evidence to confirm features are provided by the SDK to perform run-time integrity checks during execution, to verify that the security of the SDK cannot be compromised after the initialization phase by tampering with the execution code or parameters.	Run-time integrity checks are intended to ensure that only authorized libraries are used, and rogue functions are not inserted, attached or executed at run-time. One method for performing run-time integrity checks may include using hooking detection techniques specific to the underlying operating system, software development framework or language. However, other methods could be used to achieve the same objective.
	T.1.3.2 Where these checks implement the use of a hash function to validate the integrity of the 3DS SDK executable, the tester shall examine vendor materials and other evidence to confirm that the hash function meets PCI requirements of strong cryptography, including applicable cryptography requirements in this standard.	

Requirements	Assessment Procedures	Guidance
	T.1.3.3 The tester shall confirm that these checks include tests to identify attacks that aim to perform interruption of code execution or flow, interception and modification of data elements as they are processed, or modification of responses from the SDK to the calling application.	
	T.1.3.4 The tester shall determine where data values are stored (even temporarily) outside of the 3DS SDK code itself, or the memory space of the 3DS SDK provided by the device operating system during execution—e.g., written to the device file system, stored in system functions such as a "key store," etc.—and confirm that features or methods are applied to protect these values.	
	T.1.3.5 The tester shall determine where other code, data, script, or features of the application are not included in the integrity check, and confirm that having these features out of scope does not affect the security of the SDK or 3DS transaction process.	
	T.1.3.6 Based on the information provided in T.1.3.1 through T.1.3.5, the tester shall examine vendor materials and other evidence, including source code, to confirm that the claimed features are correctly implemented.	
	T.1.3.7 The tester shall test the 3DS SDK by attempting to modify the 3DS SDK prior to and during execution. Testing shall include attempts to modify the 3DS SDK code itself or values used by the code (for example, modifying configuration files, the runtime code, encryption keys, or keys or parameters stored temporarily in files or live memory during execution that could compromise the secure execution of the SDK.) The tester shall then observe the response of the 3DS SDK to confirm these modifications are detected. The changes must be made in such a way to attempt to avoid detection. Where the 3DS SDK code may be present in different locations (such as in the form of a pre-compiled file, as well as an ahead-of-time compilation that is ready to execute) the tester shall attempt to modify the 3DS SDK code in each location.	

Requirements	Assessment Procedures	Guidance
	T.1.3.8 The tester shall test the 3DS SDK by attempting to execute the 3DS SDK within an execution environment that allows for dynamic modification—such as a system that implements a hooking framework, a virtual machine (VM), or a device running a customized operating system to allow for such attacks to confirm that such modification attempts are detected by the 3DS SDK.	
1.4 Protection against Reverse Engineering The 3DS SDK binaries are protected from reverse engineering.	**T.1.4.1** The tester shall examine vendor materials and other evidence to confirm that features are provided by the 3DS SDK to protect the 3DS SDK and any data structures that may be stored in memory, the operating system file system, or other storage locations (such as an OS key store) from reverse engineering. ***Note**: This requirement is focused on the determination of the data flow and functions of the 3DS SDK, not necessarily the secrecy of the data.*	String and code obfuscation tools and techniques might be sufficient to make the reverse engineering of 3DS SDK binaries impractical depending upon the implementation. Properly implemented runtime application self-protection (RASP) and/or anti-debugging techniques could also be used.
	T.1.4.2 The tester shall determine where the SDK or data structures are not covered by these protections, and confirm this lack of protection does not affect the security of the SDK or 3DS operation.	
	T.1.4.3 The tester shall determine all locations where functions provided by the 3DS SDK are executed. This will include the main processing environment of the device, but may also include other local execution environments (such as a Trusted Execution Environment or embedded security processor).	
	T.1.4.4 Where cryptography is implemented for the purposes of obfuscation and anti-tamper, the tester shall determine the locations and data protected by those methods. The tester shall also determine what protection is provided by the cryptography (confidentiality, integrity, or both) and what algorithms and modes of operation are used. The tester shall confirm that cryptography meets PCI requirements for strong cryptography, including applicable cryptography requirements in this standard, and that all keys used for these cryptographic operations are protected.	

Requirements	Assessment Procedures	Guidance
	T.1.4.5 Where protections are provided (partially or wholly) through code obfuscation, the tester shall perform the following:	
	T.1.4.5.1 Examine vendor materials and other evidence, including application installation files where the protection methods have been applied, and compare these files to files where protections have not yet been applied to confirm the validity of the vendor attestations and documentation regarding the protection methods implemented. ***Note**: This test may require the 3DS SDK Vendor to provide both obfuscated and un-obfuscated binaries or source code to the 3DS SDK Lab to validate this requirement.*	
	T.1.4.5.2 Determine the comparative file sizes between unprotected and protected application samples, as well as the relative compression ratio of each file type when general purpose compression functions are applied to confirm that such analysis does not disclose any sensitive information about the 3DS SDK.	
	T.1.4.5.3 Examine vendor materials and other evidence, and test the software by attempting to reverse engineer the code or extract details of the code execution (e.g., through extraction of ASCII strings, functional linking/interface tables such as PLT/GOT, etc.) to confirm such attempts do not result in the disclosure of any sensitive information about the 3DS SDK.	
	T.1.4.5.4 Analyze any areas of non-traditional execution where the obfuscation relies on virtualized/interpreted commands, non-deterministic operations, or other such techniques to confirm that the exploitation of such techniques does not result in the disclosure of any sensitive information about the 3DS SDK.	

Requirements	Assessment Procedures	Guidance
	T.1.4.6 Where protections are provided by the operating environment, the tester shall perform the following:	
	T.1.4.6.1 Examine vendor materials and other evidence, including source code to confirm that such protections provide the required tamper-resistance features and that any elements of code or data that are not covered by these protections cannot be used to reverse engineer the code or disclose sensitive information about the 3DS SDK.	
	T.1.4.6.2 Test the 3DS SDK to confirm that the 3DS SDK will only execute on platforms which provide such integrated protections.	
	T.1.4.7 Where protections are provided by runtime methods or anti-debugging features, the tester shall perform the following:	
	T.1.4.7.1 Examine vendor materials and other evidence to confirm that such protections provide the required tamper-resistance features and that any elements of code or data that are not covered by these protections cannot be used to reverse engineer the code or disclose sensitive information about the 3DS SDK.	
	T.1.4.7.2 Confirm that the local software that provides these features is itself protected.	
	T.1.4.7.3 Where any features require interaction with an external system (such as a cloud-based monitoring system), the tester shall confirm that mechanisms are in place to prevent disabling of the remote protections, such as through traffic or communications manipulation.	
	T.1.4.8 Where additional protections are provided by the application, the tester shall confirm that these protections apply across all supported platforms and operating systems (as assessed under Requirement 1.1, "Security Checks"), or that any gaps that exist in coverage of these protections do not increase the risk posed by those platforms.	

Requirements	Assessment Procedures	Guidance
	T.1.4.9 Where device-specific features are relied upon, the tester shall attempt to execute the 3DS SDK on a system that either does not provide such features or has been modified to prevent the secure use of these features, and observe the operation of the 3DS SDK to confirm that the 3DS SDK does not execute when such features are absent or disabled.	
1.5 Protection of 3DS SDK Reference Data 3DS SDK Reference Data is securely stored within the 3DS SDK code to prevent unauthorized modification.	**T.1.5.1** The tester shall examine vendor materials and other evidence to identify all 3DS SDK Reference Data used or required by the 3DS SDK, which must be protected against modification (see Table 2, "Sensitive 3DS SDK Data Elements").	The 3DS SDK Version Number and 3DS SDK Reference Number are values used during 3-D Secure transactions as part of the cardholder authentication process. To properly vet the cardholder and to identify the trustworthiness of the environment in which a transaction has been created, it is important that the values are protected from unauthorized modification. Examples of methods for securely storing the 3DS SDK Reference Number or 3DS SDK Application ID (sdkAppID) in code might include obfuscation or the use of cryptography.
	T.1.5.2 The tester shall examine vendor materials and other evidence to confirm that features are provided by the 3DS SDK to protect each element of the 3DS SDK Reference Data listed above. Where there is any 3DS SDK Reference Data that is not covered by these protections, the tester shall confirm that the lack of protection does not affect the security of the 3DS SDK or 3DS transaction process.	
	T.1.5.3 Where cryptography is implemented for the purposes of providing this protection, the tester shall confirm that the cryptographic protections include the protection of the integrity of the data. The tester shall also confirm that cryptography meets PCI requirements for strong cryptography, including applicable cryptography requirements in this standard, and that all keys used for these cryptographic operations are protected.	
	T.1.5.4 Where obfuscation is implemented to provide the protections, the tester shall confirm that this obfuscation is covered under testing performed in Requirement 1.4, "Protection against Reverse Engineering."	
	T.1.5.5 Where device-specific features are relied upon to provide the protections, the tester shall attempt to execute the 3DS SDK on a system that either does not provide such features or has been modified to prevent the secure use of these features to confirm that the 3DS SDK does not execute when such features are absent or disabled.	

Requirements	Assessment Procedures	Guidance
	T.1.5.6 The tester shall attempt to circumvent the features protecting the 3DS SDK Reference Data and modify this data in such a way that the modification is not detected by the 3DS SDK upon execution to confirm the 3DS SDK prohibits such modifications. This testing must consider the different types of protections applied and devices targeted by the SDK.	

Security Objective 2: Protect Sensitive 3DS SDK Data Elements

Certain types of information collected in association with 3DS transactions are highly sensitive in nature and must be protected from unauthorized disclosure. Such information might include, but is not limited to, cardholder data (CHD), 3DS authentication data, cryptographic keys, and consumer device information. Refer to Table 2, "Sensitive 3DS SDK Data Elements," in the "Scope of Security Requirements" section for more information on which specific 3DS SDK data elements require protection from unauthorized disclosure.

Requirements	Assessment Procedures	Guidance
Requirement 2: Sensitive 3DS SDK data elements are protected from unauthorized disclosure.		
2.1 Collection of Sensitive 3DS SDK Data Elements The 3DS SDK collects and retains only the sensitive 3DS SDK data elements absolutely necessary for the software to perform its intended purpose and functionality, and only for the duration necessary.	**T.2.1.1** The tester shall examine vendor materials and other evidence to determine all sensitive 3DS SDK data elements used or required by the 3DS SDK. Vendor evidence should account for the name of the data element collected, the duration for which the data element is retained, how the data element is stored (e.g., in memory only, in the OS file system, in an OS storage mechanism such as a key store, in a device mechanism such as a Trusted Execution Environment, etc.), and how the data element is securely deleted after storage.	To ensure that the 3DS SDK does not disclose sensitive 3DS SDK data elements to unauthorized parties, the 3DS SDK should only collect the sensitive 3DS SDK data elements absolutely necessary to perform its expected functionality. Collecting sensitive 3DS SDK data elements that do not directly support the functionality of the 3DS SDK presents the opportunity for the information to be overlooked, mishandled, or otherwise insufficiently protected.
	T.2.1.2 The tester shall examine vendor evidence and other materials, including source code, to determine the functionality provided by the 3DS SDK and confirm that the functionality contained within the 3DS SDK correctly aligns with the vendor materials and evidence supplied and assessed in T.2.1.1.	
	T.2.1.3 Given the output of T.2.1.1 and T.2.1.2, the tester shall reference Table 2, "Sensitive 3DS SDK Data Elements," to confirm that the list of sensitive 3DS SDK data elements identified in T.2.1.1 is exhaustive and correct given the functionality of the 3DS SDK under evaluation. Where sensitive 3DS SDK data elements are collected that are not required for the attested functionality, the tester shall note this as a non-compliance.	

Requirements	Assessment Procedures	Guidance
	T.2.1.4 For each sensitive 3DS SDK data element identified in T.2.1.1, the tester shall determine whether the element is retained, and confirm that all sensitive 3DS SDK data elements that are retained are allowed to be retained, as noted in Table 2, "Sensitive 3DS SDK Data Elements."	
	T.2.1.5 The tester shall test the 3DS SDK by performing a series of 3DS operations, ensuring that these cover all functionality provided by the 3DS SDK, to confirm that all sensitive 3DS SDK data elements used by the 3DS SDK correctly and completely align with the sensitive 3DS SDK data elements identified in T.2.1.1. ***Note**: This testing must be performed against a 3DS test host/harness that emulates all required 3DS functionality and data elements, and allows for the monitoring of traffic to the 3DS SDK.*	
	T.2.1.6 The tester shall test the 3DS SDK by performing a series of 3DS operations to determine how the sensitive 3DS SDK data elements are stored and retained, and confirm that the use and retention of sensitive 3DS SDK data elements correctly and completely aligns with the details provided in T.2.1.1. ***Note**: This testing may be achieved through operation of the 3DS SDK in a virtualized environment that allows for monitoring the memory and storage of the system during processing, through the use of tools to monitor the data elements during operation on a physical device, or other means that will allow for confirmation of the use the memory and storage space of the 3DS SDK operating environment. It is also noted that this testing may require assistance from the 3DS SDK Vendor to disable protections in the software that would otherwise prevent the use of these types of tools.*	

Requirements	Assessment Procedures	Guidance
2.2 Clearing of Sensitive 3DS SDK Data Elements Sensitive 3DS SDK data elements collected by the 3DS SDK in association with 3DS transactions are securely deleted after 3DS transaction processing is complete and never retained, unless retention is explicitly permitted.	**T.2.2.1** Referencing the information produced in T.2.1.1, the tester shall examine vendor materials and other evidence, including source code, to confirm that each of the sensitive 3DS SDK data elements is securely deleted after use and that the methods used ensures that each sensitive 3DS SDK data element is rendered irretrievable to any subsequent process, component, functions, or applications after secure deletion.	Sensitive 3DS SDK data elements collected in conjunction with 3DS transactions should only be retained for as long as required to complete that transaction. After 3DS transaction processing is complete, any and all locations where the sensitive 3DS SDK data elements have been retained should be securely wiped or overwritten, or the sensitive 3DS SDK data elements rendered irretrievable such that any subsequent process, component, function, application, entity, etc., within the environment may not capture the information. Only in circumstances where the retention of specific sensitive 3DS SDK data elements is explicitly permitted should they be retained after 3DS transaction processing is complete. Refer to Table 2, "Sensitive 3DS SDK Data Elements," in the "Scope of Security Requirements" section for more information.
	T.2.2.2 Where secure deletion is prevented by the nature of the 3DS SDK operating environment (e.g., through virtualized memory and garbage-collection processes), the tester shall examine vendor materials and other evidence to confirm that additional protections have been implemented beyond secure deletion of the data element, and that such protections are sufficient to be considered equal to industry best practice.	
	T.2.2.3 Where additional protections or secure deletion methods are required to be implemented to compensate for lack of direct memory access in the 3DS SDK operating platform, the tester shall confirm that these methods are covered by the reverse-engineering protections tested under Requirement 1.4, "Protection against Reverse Engineering," and that any cryptography used is covered under the testing of Requirement 3.1, "Approved Algorithms and Modes of Operation."	

Requirements	Assessment Procedures	Guidance
	T.2.2.4 The tester shall test the 3DS SDK by performing a series of 3DS operations, ensuring that these cover all functionality provided by the 3DS SDK to confirm that each of the sensitive 3DS SDK data elements covered in T.2.1.1 is rendered irretrievable in accordance with the methods identified in T.2.2.1 through T.2.2.3. ***Note**: This testing must be performed against a 3DS test host/harness that provides all required 3DS functionality and data elements, and allows for the monitoring of traffic to the 3DS SDK. This testing may also require assistance from the 3DS SDK Vendor to disable protections in the software that would otherwise prevent the use of these types of tools.*	
2.3 Use of Third-Party Services The 3DS SDK uses third-party services and components only when and where it is documented and justified as part of the 3DS SDK architecture.	**T.2.3.1** The tester shall examine vendor materials and other evidence to confirm that the vendor maintains an inventory of all third-party services and components used by the 3DS SDK. **T.2.3.2** Referring to the information produced in T.2.1.1, the tester shall examine vendor materials and other evidence, including source code, to determine all sensitive 3DS SDK data elements that are passed to third-party components or services. ***Note**: Validation of this requirement must also consider whether the 3DS SDK has any advertising, machine learning, data collection, logging, tracking, or security features which rely on third-party components, features, or external services. This list of items is to be considered a minimum set and is not considered exhaustive of all potential third-party features which must be considered under this requirement.*	The use of third-party services or components should be carefully controlled and justified. Control over sensitive information may no longer reside with the 3DS SDK Vendor once sensitive information is shared or made accessible to third-party services or components, and 3DS SDK Vendors should consider the ramifications of third-party misuse or disclosure of such information.

Requirements	Assessment Procedures	Guidance
	T.2.3.3 Where third-party services are used, interfaced with, or operated by the 3DS SDK, the tester shall examine vendor materials and other evidence to confirm the vendor provides reasonable and documented justifications for the use of each third-party system or components and that the vendor maintains processes for addressing vulnerabilities in those systems or components in accordance with Requirement 4.4, "Vulnerability Identification and Monitoring."	
	T.2.3.4 The tester shall test the 3DS SDK by performing a series of 3DS operations, ensuring that these cover all functionality provided by the 3DS SDK, to determine how any third-party components or services are utilized during this operation and which data elements are sent to third parties. The tester shall confirm this correctly and completely aligns with the vendor materials and evidence provided in T.2.3.1 and T.2.3.2. ***Note**: This testing must be performed against a 3DS test host/harness that provides all required 3DS functionality and data elements, and allows for the monitoring of traffic to the 3DS SDK. This testing may also be achieved through operation of the 3DS SDK in a virtualized environment that allows for monitoring the memory and storage of the system during processing, through the use of tools to monitor the data elements during operation on a physical device, or other means that will allow for confirmation of the use of third-party components and services. It is noted that this testing may require assistance from the 3DS SDK Vendor to disable protections in the software that would otherwise prevent the use of these types of tools.*	

Requirements	Assessment Procedures	Guidance
	T.2.3.5 The tester shall test the 3DS SDK by performing a series of 3DS operations, ensuring that these cover all functionality provided by the 3DS SDK, and observe the traffic output from and received by the 3DS SDK to determine whether any of this traffic is external or extraneous to the 3DS test host to which the SDK is communicating, whether any sensitive 3DS SDK data elements are communicated through these channels, and if so, confirm that they correctly and completely align with the information provided in T.2.3.2.	
	T.2.3.6 The tester shall determine the functionality provided by the 3DS SDK during testing and confirm that this correctly and completely aligns with the information provided in T.2.3.1 to T.2.3.4.	
	T.2.3.7 The tester shall examine vendor materials and other evidence to confirm that use of third-party services is only implemented where this is a reasonably justified and documented part of the 3DS SDK architecture.	

Requirements	Assessment Procedures	Guidance
2.4 Protection against Disclosure through Unintended Channels The 3DS SDK does not disclose sensitive 3DS SDK data elements through unintended channels.	**T.2.4.1** Referring to the information produced in T.2.1.1, the tester shall examine vendor materials and other evidence, including source code, to determine how each of the data elements is generated/input and displayed (if displayed).	Proactive measures to ensure that sensitive 3DS SDK data elements are not inadvertently "leaked" should be implemented by the 3DS SDK Vendor or within the 3DS SDK. Disclosure of sensitive 3DS SDK data elements to unauthorized parties often occurs via unknown or unintended outputs or channels. For example, sensitive 3DS SDK data elements could be unintentionally disclosed through error- or exception-handling routines, logging or debugging channels, third-party services or components, or the use of shared resources such as memory, disk, files, keyboards, displays, and functions. Protective mechanisms, whether process or programmatic in nature, should be implemented to ensure that sensitive 3DS SDK data elements are not accidentally disclosed through such means. Example implementations of data leakage protection controls can be found in the *EMV® 3DS SDK Technical Guide*.
	T.2.4.2 Referring to the information produced in T.2.1.1 and the details generated above, the tester shall confirm that for each sensitive 3DS SDK data element identified in T.2.1.1, the vendor has implemented protections to safeguard that data element against disclosure through unintended channels.	
	T.2.4.3 Where the sensitive 3DS SDK data element is input by the cardholder, the tester shall confirm that methods are implemented by the 3DS SDK to mitigate clickjacking, screen overlay, or other such input-stealing attacks.	
	T.2.4.4 For all sensitive 3DS SDK data elements identified in T.2.1.1, the tester shall confirm that methods are implemented by the 3DS SDK to mitigate capture of each of these elements through use of shared resources such as memory or file systems.	
	T.2.4.5 Referring to testing performed in Requirement 2.3, "Use of Third-Party Services," the tester shall confirm that methods are implemented to mitigate the capture or exposure of each sensitive 3DS SDK data element as it is passed between the 3DS SDK and any third-party services or components.	
	T.2.4.6 Referring to the information produced in T.2.1.1, the tester shall examine vendor materials and other evidence, including source code, to confirm that only sensitive 3DS SDK data elements that are explicitly permitted to be hard-coded are stored in the source code.	

Requirements	Assessment Procedures	Guidance
	T.2.4.7 The tester shall examine source code to determine whether sensitive 3DS SDK data elements which are externally generated or provided are processed in a way that indicates they are static—for example, where they utilize a third-party service or component, covered under Requirement 2.3, "Use of Third-Party Services," which implements static values; or where the 3DS SDK processing clearly does not accommodate for the expected range of values which may be provided in any particular data element. In such cases, the tester shall confirm that these values are not static, and that any such attestations from the vendor are documented.	
	T.2.4.8 The tester shall examine vendor materials and other evidence, including source code, to identify all error, debugging, or other output functionality. Where such functionality is found, the tester shall confirm that the functionality does not result in the unintended disclosure or leakage of any sensitive 3DS SDK data elements.	
	T.2.4.9 The tester shall examine vendor materials and other evidence, including source code, to confirm that any functionality that results in the output of sensitive 3DS SDK data elements is intended. The tester is expected to cross reference any output functionality to the testing performed in Requirement 2.3, "Use of Third-Party Services," to validate that all communication of sensitive 3DS SDK data elements is intended.	

Requirements	Assessment Procedures	Guidance
	T.2.4.10 The tester shall test the 3DS SDK by performing a series of 3DS operations, ensuring that these cover all functionality provided by the 3DS SDK, and confirm that sensitive 3DS SDK data elements are not disclosed through unintended channels. ***Note**: This testing must be performed against a 3DS test host/harness that provides all required 3DS functionality and data elements, and allows for the use and monitoring of shared resources such as memory, keyboards and displays. The test harness must additionally allow for the capture of any error or debug data output from the 3DS SDK.*	
	T.2.4.10.1 The tester shall test the 3DS SDK by attempting to capture or otherwise determine the values of sensitive 3DS SDK data elements generated, input, or processed by the 3DS SDK. The tester must attempt methods that include both on-device capture, as well as capture through monitoring of communication channels. Communication channel capture shall consider the application of traffic analysis to determine the sensitive 3DS SDK data elements communicated.	
	T.2.4.10.2 The tester shall attempt to capture or otherwise determine the values of sensitive 3DS SDK data elements generated, input, or processed by the 3DS SDK through capture and analysis of error codes or use of debugging/test features. The tester must attempt methods that utilize both normal and forced error flows of the processing, and determine whether any sensitive 3DS SDK data elements are leaked.	

Requirements	Assessment Procedures	Guidance
2.5 Hardcoded 3DS SDK Data Elements Sensitive 3DS SDK data elements are not hard-coded in 3DS SDK code unless explicitly permitted.	**T.2.5.1** Referring to testing performed in Requirement 2.4, "Protection against Disclosure through Unintended Channels," the tester shall confirm that sensitive 3DS SDK data elements are not hardcoded in the 3DS SDK except where the vendor has maintained reasonable and documented justification for their use.	The 3DS SDK, as part of its normal functionality, will be exposed to and handle various sensitive 3DS data elements. For example, the directoryServerIDs public keys will be issued after certification and stored by the 3DS SDK. It is fairly trivial to reverse-engineer mobile applications (for example, using dex2jar or JAD) and perform analysis on the source code itself with intent to harvest hard-coded sensitive information. To prevent that, the 3DS SDK should not store any sensitive 3DS SDK data elements in the source code unless explicitly permitted. Instead—in the case of cryptographic keys, for example—the 3DS SDK could fetch the data from an HSM, then store the keys locally utilizing the most secure storage options (for example, keychain, key store, or shared preferences) provided by the operating system where appropriate. Refer to Table 2, "Sensitive 3DS SDK Data Elements," in the "Scope of Security Requirements" section for more information on which sensitive 3DS SDK data elements are permitted to be retained.
	T.2.5.2 The tester shall test the 3DS SDK by performing a series of 3DS operations, ensuring that these cover all functionality provided by the 3DS SDK, and observe the use of sensitive 3DS SDK data elements across multiple operations and executions of the 3DS SDK. Where sensitive 3DS SDK data elements appear to have the same value or a limited range of values, the tester shall confirm that these values correctly and completely align with those values noted in T.2.5.1. ***Note**: This testing must be performed against a 3DS test host/harness that has been confirmed to provide all required 3DS functionality and data elements.*	
2.6 Run-Time Data Protection The 3DS SDK implements run-time data protection techniques to protect the 3DS SDK instance from being accessed by unauthorized third-party applications and/or libraries.	**T.2.6.1** Referencing the sensitive 3DS SDK data elements identified in T.2.1.1 and the protection features determined through other testing, the tester shall confirm that protections against extraction or determination are provided for each sensitive 3DS SDK data element.	Code injection, code reuse, local and remote hooks, reverse-engineering attacks and side-channel attacks (for example, cache side-channel or timing attack) are often used to execute code in the context of target process or to extract sensitive information from the target systems and applications. Various defense techniques exist to make attacks significantly harder, including dynamic or artificial software diversity, compression and randomization, etc. Properly implemented runtime application self-protection (RASP) and/or anti-debugging or anti-hooking techniques may be used to satisfy this requirement.

Requirements	Assessment Procedures	Guidance
	T.2.6.2 The tester shall examine vendor materials and other evidence, including source code, and test the 3DS SDK to determine what sensitive 3DS SDK data elements may be most susceptible to side-channel attacks, such as cache timing or other attack methods, and to confirm that such attacks are not feasible given the implemented protections.	
	T.2.6.3 The tester shall examine vendor materials and other evidence, including source code, and test the software to determine what sensitive 3DS SDK data elements may be most susceptible to exposure through code injection or code reuse attacks, and to confirm such attacks are not feasible given implemented protections.	
	T.2.6.4 The tester shall examine vendor materials and other evidence, including source code, and test the 3DS SDK to determine what sensitive 3DS SDK data elements may be most susceptible to exposure through hooking methods (remote and local) and reverse-engineering attacks, and to confirm that such attacks are not feasible given other protections.	
	T.2.6.5 The tester shall test the 3DS SDK by attempting to subvert any third-party components or services relied upon by the 3DS SDK to determine whether any sensitive 3DS SDK data elements are used by the 3DS SDK that are not already confirmed to be passed to that third-party component or service as per testing under Requirement 2.3, "Use of Third-Party Services". Where third-party components or services are known to receive sensitive 3DS SDK data elements, the tester shall attempt to extract the sensitive values from these services during operation of the 3DS SDK to confirm the sensitive 3DS SDK data elements are not exposed to extraction or determination through code injection, code reuse, reverse engineering, and the use of hooking (remote or local) methods.	

Requirements	Assessment Procedures	Guidance
2.7 UI Protection The user interface (UI) rendered by the 3DS SDK (both in Native and HTML modes) is isolated and secured such that user information (including authentication data) displayed and captured by the 3DS SDK is not accessible to any unauthorized process outside the 3DS SDK.	**T.2.7.1** The tester shall examine vendor materials and other evidence, including source code, to identify all OS functions or other third-party features the 3DS SDK relies upon for the passing and rendering of UI elements.	UI isolation is necessary to ensure that sensitive 3DS SDK data elements cannot be leaked to any unauthorized component or process outside the 3DS SDK. Sensitive 3DS SDK data elements contained in the UI components rendered by the 3DS SDK can be inadvertently accessible to unauthorized components and processes unless adequately protected. Protection methods include restricting access to the UI components as well as controlling inter-process communications and event handling. For example, when an application goes into the background, it could prevent screen capturing by the underlying OS by intercepting the applicationDidEnterBackground event to control what information is shared with iOS—e.g., splash screen or blurred window snapshot—or preventing manual and automatic screenshots altogether by setting FLAG_SECURE on Android.
	T.2.7.2 The tester shall examine vendor materials and other evidence, including source code, to confirm that security features are implemented to protect the UI against access by other applications.	
	T.2.7.3 The tester shall test the 3DS SDK by attempting to modify, capture, or otherwise undermine the security of the 3DS SDK UI to confirm that any user information captured and displayed by the 3DS SDK is not accessible to any unauthorized process outside of the 3DS SDK.	
2.8 HTML Rendering The 3DS SDK intercepts all external URL requests made by the HTML UI rendered (both during loading of the UI and on user action) and handles these requests within the 3DS SDK. Such requests are not passed to the device's operating system or the Internet.	**T.2.8.1** The tester shall examine vendor materials and other evidence, including source code, and the findings in T.2.7.1 to confirm that URL requests made by the UI in HTML mode are handled within the 3DS SDK itself and are not passed to the device's operating system or any other component (internal or external).	When the 3DS SDK makes API calls to the ACS that are rendered in HTML mode, those calls, as well as the responses, should not be available outside the 3DS SDK. HTML content generated by the ACS and displayed in HTML mode by the 3DS SDK should not reference content from other external sites. The intent of this requirement is to reduce the 3DS SDK's attack profile and to protect against the inadvertent leakage of sensitive 3DS SDK data elements to unauthorized parties.
	T.2.8.2 The tester shall examine vendor materials and other evidence, including source code, to determine what web elements the 3DS SDK is configured to handle, and to confirm that these methods are created and used in a way that mitigates attacks and prevents references to external content that is not supplied by the Access Control Server (ACS).	

Requirements	Assessment Procedures	Guidance
	T.2.8.3 Using the information determined in T.2.8.2, the tester shall test the 3DS SDK by attempting to inject HTML references in ACS response(s), and observe the operation of the 3DS SDK to confirm that the UI processes running in HTML mode are handled by the 3DS SDK and are not passed to the device operating system or other component(s) (internal or external). ***Note**: This testing must be performed with a test host/harness that allows for such injection.*	
2.9 Prevention of External Code or Script Execution The 3DS SDK prevents the injection and execution of any JavaScript code by the HTML UI or any other process outside the 3DS SDK.	**T.2.9.1** The tester shall examine vendor materials and other evidence to confirm that protections are provided by 3DS SDK to prevent the injection and execution of JavaScript into the UI (in HTML mode) or any other process outside of the 3DS SDK.	Often an attack will use JavaScript to manipulate the functionality of an application. To prevent such attacks, the injection and execution of JavaScript should be prohibited by properly initializing the HTML UI object.
	T.2.9.2 The tester shall examine vendor materials and other evidence, including source code, to confirm that the source code does not contain any JavaScript parsing or execution code, even if disabled, and that the functionality provided in the source code for preventing the injection and execution JavaScript into the UI or other processes outside of the 3DS SDK aligns with the details provided in T.2.9.1.	
	T.2.9.3 The tester shall test the 3DS SDK by attempting to inject JavaScript into the ACS response(s), and observe the response of the 3DS SDK to confirm that this is not executed by the 3DS SDK. ***Note**: This testing must use a test host/harness that allows for this modification of the ACS responses.*	

Security Objective 3: Use Cryptography Appropriately and Correctly

The proper implementation of cryptographic algorithms and appropriate key-management techniques is critical to ensuring the integrity and confidentiality of the 3DS SDK and the information it handles. If cryptographic algorithms and key-management techniques are not implemented properly or appropriately, the value they provide from a security perspective is significantly reduced—if not eliminated altogether. Only industry-recognized algorithms and key-management techniques must be used. Proprietary algorithms are not allowed.

Requirements	Assessment Procedures	Guidance
Requirement 3: The 3DS SDK only utilizes strong cryptography and ensures that cryptographic methods are properly and appropriately implemented.		
3.1 Approved Algorithms and Modes of Operation Only approved cryptographic algorithms and methods are used. Approved cryptographic algorithms and methods are those specified within the *EMV® 3-D Secure SDK Specification*. Approved cryptographic algorithms and methods are also recognized by industry-accepted standards bodies—for example: NIST, ANSI, ISO, EMVCo, etc. Cryptographic algorithms and parameters that are known to be vulnerable are not used.	**T.3.1.1** The tester shall examine vendor materials and other evidence, including source code, to determine what cryptographic algorithms and methods are used and all cryptographic keys used in the system that are relied upon for the security of the 3DS SDK.	To protect sensitive information, the 3DS SDK should utilize only recognized cryptographic implementations based on the *EMV® 3-D Secure SDK Specification* and industry-accepted standards. For a list of approved cryptographic algorithms and methods, please refer to the *EMV® 3-D Secure SDK Specification* and the *EMV® 3-D Secure Protocol and Core Functions Specification*.
	T.3.1.2 The tester shall examine vendor materials and other evidence, including source code, to identify modes of operation available for each key, including determining how any additional values (such as initial vectors) may be generated for that mode of operation.	
	T.3.1.3 Where the mode of operation may be open to exploitation—e.g., relocation or data analysis attacks on Electronic Code Book (ECB) mode—the tester shall confirm that this sort of attack is not feasible for this implementation. This testing must always be performed for keys that allow for ECB as a mode of operation.	
	T.3.1.4 Where the mode of operation requires the use of another value, such as an Initialization Vector (IV) or counter, the tester shall confirm that the implementation ensures that this value is correct and secure.	

Requirements	Assessment Procedures	Guidance
	T.3.1.5 The tester shall examine vendor materials and other evidence, including source code, to determine all key generation or key agreement processes that are used by the system, and to confirm that they ensure keys are generated with full entropy (e.g., a 128-bit key is generated with 128 bits of entropy input).	
	T.3.1.6 The tester shall confirm that no reversible key-calculation modes (such as key variants) are used to directly create new keys from an existing key. All key-generation functions must implement one-way functions or other irreversible key-generation processes.	
	T.3.1.7 The tester shall confirm that any key signature or fingerprint values returned by the system do not reveal any details about the key itself. Key checksum values (KCVs) must be limited to five bytes or less than half of the algorithm block size, whichever is smaller, and hash algorithms used for key fingerprints (on secret or private keys) must implement SHA256 or above.	
	T.3.1.8 The tester shall confirm that a cryptoperiod is defined for each key, and that update procedures are also defined to replace each key at the end of this cryptoperiod.	
	T.3.1.9 The tester shall confirm that security is not provided to any key by a key of lesser strength—e.g., by encrypting a 256-bit AES key with a 128-bit AES key.	

Requirements	Assessment Procedures	Guidance
	T.3.1.10 For any public keys used by the system, the tester shall confirm that the authenticity of each public key is maintained. Use of public keys that are not signed or MAC'd or are maintained in self-signed certificates, is prohibited unless the authenticity of the key is ensured through use of a secure cryptographic module. Self-signed certificates that exist as part of the base platform on which the 3DS SDK is executed are excluded from this requirement.	
	T.3.1.11 The tester shall confirm that key purpose and integrity is ensured for all keys used in the system, preventing a key of one purpose (e.g., key encryption) from being replaced with a key of another purpose (e.g., general data encryption).	
	T.3.1.12 The tester shall confirm that each key has a single unique purpose, and that no keys are used for multiple purposes (such as both signing and encrypting data), and that keys used to encrypt Cardholder Verification Method (CVM) data are not used for any other operation (such as general-purpose data encryption, monitor message encryption, etc.).	
	T.3.1.13 The tester shall confirm that keys used to validate the authenticity of a datagram are unique to each endpoint, so that a (H)MAC or signature generated at one end would always be different if generated by the other end point.	

Requirements	Assessment Procedures	Guidance
3.2 Random Number Generator(s) All random numbers used by the 3DS SDK are generated using only approved random number generation (RNG) algorithms or libraries. Approved RNG algorithms or libraries are those meeting industry standards for sufficient unpredictability (e.g., *NIST Special Publication 800-22*). ***Note**: Proof that RNG algorithms or libraries meet industry standards may include recognition by industry bodies, or evidence to show where those RNG algorithms or libraries were assessed to ensure that the random numbers generated are sufficiently unpredictable.*	**T.3.2.1** The tester shall examine vendor materials and other evidence, including source code, to determine the implementation of all random number generation functions used in the 3DS SDK implementation. **T.3.2.2** The tester shall examine vendor materials and other evidence, including source code, to determine all functions of the 3DS SDK that rely upon the on-device generation of random numbers. This should include uses such as random values required in secure communications channels (such as TLS).	Random numbers are used in numerous software applications, including cryptography, to protect sensitive information. Encryption keys, initialization values (seeds), and 3DS SDK transaction IDs are examples of random numbers used in the 3DS SDK. It is not a trivial endeavor to design and implement a secure random number generator. 3DS SDK Vendors are required to use only approved random number generation algorithms and libraries, or provide evidence to illustrate how the random number generation algorithms and libraries were tested to confirm that random numbers generated are sufficiently unpredictable. The implementation may rely on either a validated cryptographic library or module (for example, Validated FIPS 140-2 Cryptographic Modules). The Vendor should have a good understanding of the installation, initialization, configuration and usage—for example, initial seeding of the random function—of the RNG mechanisms to ensure that the implementation can meet the effective security strength required for the intended use. The calls to these libraries should also be protected from being hooked.

Requirements	Assessment Procedures	Guidance
	T.3.2.3 The tester shall confirm that the 3DS SDK does not rely solely on any on-device random number generators and always uses an RNG provided by or within the 3DS SDK for the purposes of generating random values that are relied upon for the secure functionality of the 3DS SDK. The tester shall reference the random values required by the 3DS SDK listed in T.3.2.2. Where any values are generated without the use of the 3DS SDK RNG, the tester shall confirm the use of the RNG is prevented by the platform targeted by the 3DS SDK, and that the use of the on-platform RNG does not violate the security of the 3DS operations.	
	T.3.2.4 The tester shall confirm that values provided by the RNG are sufficiently random in accordance with Requirement 3.3, "Random Number Entropy."	
	T.3.2.5 The tester shall examine vendor materials and other evidence to determine any requirements for the developer integrating the 3DS SDK to ensure that the random numbers are sufficiently random. The tester shall confirm that there is clear and sufficient guidance outlining these requirements made available to stakeholders in accordance with Requirement 5.1, "Availability of Stakeholder Guidance."	
3.3 Random Number Entropy Random values have entropy that meets the minimum effective security strength requirements of the cryptographic primitives and keys that rely on them.	**T.3.3.1** The tester shall examine vendor materials and other evidence, including source code, and the results of testing performed in Requirement 3.2, "Random Number Generator(s)," to determine how the RNG within the 3DS SDK is implemented and how the entropy for the RNG is generated.	Note that a non-deterministic random number generator (NDRG) may produce an output string that contains less entropy than implied by the length of the output. A deterministic random number generator (DRNG) is dependent on the entropy of its seed value. Vendors are encouraged to use as many sources of seed material as possible to ensure random number values are sufficiently random.
	T.3.3.2 Where the 3DS SDK relies upon an RNG that has been approved under the NIST Cryptographic Algorithm Validation Program (CAVP), the tester shall confirm from the approval and/or security policy of the RNG, whether the RNG requires the initial entropy to be seeded externally.	

Requirements	Assessment Procedures	Guidance
	T.3.3.3 Where the 3DS SDK is required to generate entropy through use of its own RNG or a RNG that requires external seeding, the tester shall confirm that there is sufficient entropy generated—e.g., through confirmation that the entropy generation involves inputs that cannot be predicted within the domain of the random values produced by the RNG.	
	T.3.3.4 The tester shall confirm that the RNG is seeded with a random value of at least 256 bits for use during all operations.	
	T.3.3.5 The tester shall obtain at least two sets of 64MB of random data from each of the RNG implementations used in the system, generated during separate installs and initial executions on the same device. This data may be supplied directly by the vendor, but the tester must detail the method used to generate this data, and justify why this sufficiently replicates the way in which the RNG will be used by the system after two similar installations. The tester shall combine the two sets of data and pass this 128MB of data through the NIST STS test program, and detail the results, indicating pass and fail results and how these demonstrate compliance to this requirement. In some situations, it is necessary to repeat such tests using additionally obtained data to confirm final results.	

3DS SDK Vendor Security Requirements and Assessment Procedures

Security Objective 4: Manage Risks and Vulnerabilities

A critical aspect of secure software is the software vendor's ability to identify and resolve vulnerabilities before they can be exploited. To be effective at this, the software vendor must have well-structured and repeatable processes in place to identify potential risks, to design and implement security controls to address those risks, and to resolve any vulnerabilities that are identified post-release. The implemented processes should include analyzing any risks associated with the use of third-party components or services, preventing the reintroduction of previously addressed vulnerabilities, and should result in the timely release of security updates.

Requirements	Assessment Procedures	Guidance
Requirement 4: Risks and vulnerabilities affecting the 3DS SDK are minimized and addressed in a timely manner.		
4.1 Threat and Vulnerability Analysis Threats, attack scenarios and/or attack vectors applicable to the 3DS SDK are known, analyzed, documented, and described in terms of their exploitability, impact, and residual risk.	**T.4.1.1** The tester shall examine vendor materials and other evidence to confirm that a process is implemented by the 3DS SDK Vendor for identifying, documenting, and analyzing threats, vectors, and attack scenarios applicable to the 3DS SDK.	The design of the 3DS SDK should be evaluated to identify common attack scenarios and/or potential attack vectors applicable to the 3DS SDK, and the results of that analysis documented. Documentation should describe the various aspects of the code that could be attacked (including things that frameworks and libraries do on behalf of the 3DS SDK), the difficulty in mounting a successful attack, how widely the program will be distributed, what mitigation techniques are used (for example, how the security functionality of the operating system is leveraged), and identify or define a methodology for measuring the likelihood and impact of an exploit. Those individuals making the residual risk determinations should be independent of those individuals responsible for the development of the 3DS SDK. The need for independence is to ensure that only disinterested individuals make assessments and not individuals with objectives that may be in conflict with security concerns.
	T.4.1.2 The tester shall confirm that the process required is sufficiently detailed for it to be repeatable across different personnel and locations.	
	T.4.1.3 The tester shall confirm that the process clearly outlines the individuals or teams responsible for determining and investigating new threats. It is acceptable if a group or job title is referenced, but the tester must ensure that there is a clear line of responsibility for this item.	
	T.4.1.4 The tester shall interview a sample of the personnel identified in T.4.1.3 and confirm that they are aware of the policy and procedure requirements for the analysis of new threats. The tester shall also examine vendor materials and other evidence produced by these interviewees to confirm that defined processes are being followed.	

Requirements	Assessment Procedures	Guidance
	T.4.1.5 The tester shall confirm from the evidence identified in T.4.1.1 that methods are defined and used for categorizing and ranking threats. The tester shall confirm that a documented methodology exists, which can be reasonably assumed to produce the same results each time it is enacted (assuming the same threat and threat environment). It is not a requirement that a public ranking method is used; it is acceptable for the vendor to implement its own method if this provides sufficient assurance and repeatability.	
	T.4.1.6 The tester shall interview a sample of the personnel identified in T.4.1.3 and confirm that they understand and can apply the categorizing and ranking methodology employed by the vendor.	
	T.4.1.7 For a sample of threats identified in T.4.1.4, the tester shall obtain the categorizing and ranking results for the sample and confirm that they align with the documented process.	
4.2 Development of Defensive Strategies Defensive strategies and mechanisms to protect against attack vectors and/or scenarios are designed and implemented. Attack scenarios that are applicable to the 3DS SDK but are not specifically addressed are justified.	**T.4.2.1** The tester shall examine vendor materials and other evidence to confirm that there are clear, documented vendor policy and procedure statements regarding the remediation of identified vulnerabilities in the 3DS SDK. These statements must tie together with the identification and ranking process covered under Requirement 4.1, "Threat and Vulnerability Analysis."	Once threats, attack vectors, and attack scenarios are identified, they should be mitigated. 3DS SDK Vendors should define and implement mechanisms to protect the 3DS SDK from those risks and reduce the likelihood and impact of their exploitation. Any known risks that are not addressed or do not reduce the likelihood and impact of the exploitation of those risks to a reasonable level should be justified.
	T.4.2.2 The tester shall determine whether the vendor explicitly allows for potential threats to remain un-addressed and, if so, the tester shall confirm that ranking/categorization levels are considered acceptable for this (as assessed in Requirement 4.1, "Threat and Vulnerability Analysis"), and that either this ranking process or another process explicitly involves a step to document and justify why it is acceptable to not address this vulnerability specifically.	

Requirements	Assessment Procedures	Guidance
	T.4.2.3 The tester shall interview personnel responsible for the implementation of defensive strategies and confirm that they know of and understand the policy and procedure requirements for this process.	
	T.4.2.4 Referencing the documented threats and vulnerabilities sampled in Requirement 4.1, "Threat and Vulnerability Analysis," the tester shall determine whether any vulnerabilities have been not specifically remediated and, if so, confirm that this is due to the correct and documented steps involved in the policy and procedures identified in T.4.2.1. Where all vulnerabilities have been addressed, the tester shall obtain more evidence to address this testing requirement. If vendor policy is to mitigate all threats and vulnerabilities, the tester shall require an increased sample size to confirm that each and every threat has been addressed.	
4.3 Software Security Testing Software security testing is an integral part of the 3DS SDK's life cycle and is performed throughout the software life cycle to confirm that risks and attack scenarios are addressed, defensive mechanisms are implemented properly, and the propagation of design flaws or vulnerabilities into production code is prevented.	**T.4.3.1** The tester shall examine vendor materials and other evidence to confirm that the vendor has written policy and procedures requiring internal security review and testing that accounts for the entire 3DS SDK code base, including detecting vulnerabilities in code developed by the vendor, as well as vulnerabilities in third-party, open source, or shared components or libraries.	Software security testing is a fundamental practice to ensure that software cannot be exploited through known vulnerabilities or common attacker techniques. Performing security testing throughout the development process and during development of future updates using a variety of testing techniques mitigates the risk that vulnerabilities may be introduced during updates. Testing tools and techniques may include manual code reviews, static code analysis, dynamic code analysis, software composition analysis, fuzz testing, penetration testing, etc., where appropriate. Organizations are responsible for understanding common vulnerabilities associated with the technologies they are using and for implementing testing practices specific to addressing those vulnerabilities.
	T.4.3.2 The tester shall confirm that the process for testing of internal code involves both manual and automated means.	
	T.4.3.3 The tester shall confirm that the process clearly outlines the individuals or teams responsible for this testing. It is acceptable if a group or job title is referenced, but the tester must ensure that there is a clear line of responsibility for this item.	

Requirements	Assessment Procedures	Guidance
	T.4.3.4 The tester shall confirm that the process includes ensuring that the processes for identification and mitigation of threats are correctly performed prior to the release of any production code.	
	T.4.3.5 The tester shall confirm that the process includes ensuring that any test, debug, or other code that is intended only for internal use is removed prior to release to production.	
	T.4.3.6 Referencing threats sampled in the tests for Requirement 4.1, "Threat and Vulnerability Analysis," the tester shall examine vendor materials and other evidence, interview personnel, and test the 3DS SDK to confirm that threats identified and noted as required to be mitigated were addressed before the 3DS SDK was released.	

Requirements	Assessment Procedures	Guidance
4.4 Vulnerability Identification and Monitoring The 3DS SDK and its components are monitored for vulnerabilities. In addition to their own processes, 3DS SDK Vendors provide mechanisms to enable third parties to report vulnerabilities. Information on identified vulnerabilities is maintained. Vulnerabilities are addressed and software updates are made available to all stakeholders in a timely manner. All exceptions are documented and justified. The reoccurrence of previously addressed vulnerabilities is tracked and minimized.	**T.4.4.1** The tester shall examine vendor materials and other evidence to confirm that there is a documented release policy for the 3DS SDK, and that this ensures the processes outlined in previous 4.x requirements are followed before the SDK is released to production.	The identification and management of vulnerabilities in the 3DS SDK and its components is not a one-time exercise. New vulnerabilities can be introduced at any time; therefore, it is imperative that the 3DS SDK is continuously monitored for vulnerabilities and appropriate action is taken when vulnerabilities are identified. 3DS SDK Vendors should have processes in place to identify vulnerabilities in the 3DS SDK and its components, including vulnerabilities identified by third parties, and resolve those vulnerabilities in a timely manner such that the integrity, confidentiality, and the overall confidence in the security of the 3DS SDK are maintained. It is understood that some vulnerabilities may not pose a risk to the application software or environment. However, regardless of the criticality or the impact of the vulnerability, it is important that all vulnerabilities are identified, their risk is known and that there is a process that recognizes, evaluates and (if necessary), assumes the risk. The process should include management review and approval.
	T.4.4.2 The tester shall confirm that the release policy clearly outlines the acceptable period of time after which patches are made available for the different rankings of vulnerabilities as defined in previous 4.x requirements.	
	T.4.4.3 The tester shall examine vendor materials and other evidence, and interview personnel to confirm that the vendor has an explicit procedure in place for the acceptance and processing of new vulnerabilities through external communications. Although not mandated, this requirement can be met by a properly administered bug bounty program. It does require that reported vulnerabilities are formally registered and processed according to the documented process previously assessed in the 4.x requirements.	

Requirements	Assessment Procedures	Guidance
	T.4.4.4 The tester shall examine vendor materials and other evidence, and interview personnel to confirm that there is a public-facing procedure for the reporting of vulnerabilities in the 3DS SDK. This procedure must implement methods to ensure the confidentiality of the vulnerability as it is reported. For example, a process that requires the reporting of a vulnerability to a shared "info@[company]" e-mail address, without additional encryption, would be non-compliant to this requirement. ***Note**: Use of a specific web portal secured with TLS (using acceptable ciphersuites), and/or e-mails secured with strong cryptography are examples of acceptable methods to secure the confidentiality of vulnerability reporting.*	
	T.4.4.5 The tester shall examine vendor materials and other evidence, and interview personnel to confirm that, where any such third-party vulnerability reports have been accepted and processed, the process appears correct—e.g., through validation of any special e-mail address or portal that is to be used for public vulnerability reporting.	
	T.4.4.6 The tester shall examine vendor materials and other evidence to confirm that there is a process to validate that new releases have not re-introduced older vulnerabilities. This may involve the process being updated to specifically check for vulnerabilities as they are discovered, or ensuring that older and un-patched software components and libraries are removed from the development environment as they are updated.	

Requirements	Assessment Procedures	Guidance
	T.4.4.7 The tester shall examine vendor materials and other evidence, and interview personnel to confirm that the process as documented is understood and followed. Where previously sampled vulnerabilities identified the need for an update to internal libraries or components, the tester shall confirm through these interviews and evidence that these have been correctly updated and the older, unpatched versions have been removed.	
	T.4.4.8 Where not covered under previous requirements, the tester shall examine vendor materials and other evidence, and interview personnel to confirm that any decisions not to address vulnerabilities are reasonably justified and documented.	
4.5 Updates During Transaction Processing The 3DS SDK Vendor does not enable updates or changes to 3DS SDK functionality to be pushed to the 3DS SDK during 3DS transaction processing.	**T.4.5.1** The tester shall examine vendor materials and other evidence to confirm that the 3DS SDK does not accept updates during 3DS transaction processing.	Updates to 3DS SDK functionality during 3DS transaction processing should not be permitted to ensure the integrity of those transactions. 3DS SDK Vendors should implement functionality within the 3DS SDK to preserve the integrity of the 3DS SDK code during 3DS transaction processing, while still enabling the 3DS Requestor App to push automatic updates to the consumer device when necessary.
	T.4.5.2 The tester shall examine vendor materials and other evidence, including source code, to confirm that methods are implemented to prevent the SDK from being updated during 3DS transaction processing.	
	T.4.5.3 Where the operating systems and platforms targeted by the 3DS SDK do not allow for the applications to prevent or delay updates themselves, the tester shall confirm that other protections have been put in place by the vendor to mitigate interruption of the 3DS transaction process during updates.	
	T.4.5.4 The tester shall test the 3DS SDK by performing a series of 3DS transactions within a test host/harness that allows for updates to be pushed to the 3DS SDK, to confirm that the 3DS SDK does not accept updates during the transaction processing, and that the outcome of this testing aligns with the vendor materials and evidence provided in T.4.5.1 through T.4.5.3.	

Security Objective 5: Provide Guidance to Stakeholders

3DS SDK Vendors have a responsibility to help facilitate the secure implementation of their 3DS SDK(s). While 3DS SDK Vendors may not be actively involved in the implementation of their product(s), they are capable of providing instructions to stakeholders on how to implement and configure the 3DS SDK as securely as possible. Without vendor guidance, 3DS SDK functionality is likely to be implemented into 3DS Requestor applications improperly. Insecure implementation of 3DS SDKs could create vulnerabilities within an embedding 3DS Requestor App that could result in significant downstream effects.

Requirements	Assessment Procedures	Guidance
Requirement 5: The 3DS SDK Vendor provides written security guidance to stakeholders.		
5.1 Availability of Stakeholder Guidance The 3DS SDK Vendor creates, maintains, and makes available guidance to all stakeholders on the appropriate and secure implementation, configuration, and use of the 3DS SDK as well as all APIs provided by the 3DS SDK.	**T.5.1.1** The tester shall examine vendor materials and other evidence to confirm that the 3DS SDK Vendor maintains detailed security guidance for the secure implementation of the 3DS SDK, as determined in previous testing within this standard, and that such guidance contains all references required for a secure implementation and configuration of the 3DS SDK.	Detailed implementation and security guidance for stakeholders helps to direct stakeholders and integrators during the implementation of the 3DS SDK into a Requestor App. Without detailed vendor security guidance, appropriate configuration and use of the 3DS SDK could be overlooked and unknowingly left out of the 3DS SDK security controls, thus leaving the device vulnerable to compromise.
	T.5.1.2 The tester shall confirm that vendor security guidance is made available to all software developers who will be integrating the 3DS SDK into their applications. The tester shall also confirm there are no specific legal, distribution, or other requirements that appear to prevent the distribution of the security guidance to developers who require this guidance—e.g., a data classification that prevents the document from being distributed to other entities.	
	T.5.1.3 The tester shall confirm that the security guidance identifies all configurable security-related options and parameters of the 3DS SDK, and provides guidance on how to properly configure and secure these options and parameters.	

Requirements	Assessment Procedures	Guidance
	T.5.1.4 For all scenarios where the 3DS SDK receives or generates sensitive 3DS SDK data elements, the tester shall confirm that the security guidance specifically notes how these are to be transmitted to/from the 3DS SDK in a secure manner. The tester shall reference testing performed under Requirement 1 to confirm the correct guidance for all sensitive 3DS SDK data elements used.	
	T.5.1.5 Where the 3DS SDK requires entropy input from the application for the purposes of seeding the random number generator, the tester shall confirm that the security guidance includes examples of methods on how to successfully generate entropy on the end system, and how much entropy is required for the secure operation of the 3DS SDK.	
	T.5.1.6 The tester shall confirm that the vendor has a documented policy and procedure for the generation of the security guidance prior to release of the 3DS SDK.	
	T.5.1.7 The tester shall confirm that an individual or group is assigned the clear responsibility for the maintenance and update of the security guidance. The tester shall interview a sample of these individuals and confirm they understand the requirements for the security guidance, and that they are aware of their responsibility for managing this information.	

Requirements	Assessment Procedures	Guidance
5.2 Disclosure of Updates to Stakeholders Upon any changes and/or updates to the 3DS SDK, the 3DS SDK Vendor provides all stakeholders with a clear and unambiguous indication that updates were made, and makes available detailed information regarding the specific changes made, the functionalities that are affected, and the potential security impacts associated with those changes (for example, release notes). The relationship between the updated software and detailed information about the updates should be clear.	**T.5.2.1** The tester shall examine vendor materials and other evidence to confirm that procedures exist to clearly communicate changes to the 3DS SDK security guidance to relevant stakeholders. This process must require the initiation of contact regarding the change from the 3DS SDK Vendor; it is not considered acceptable to post changes on a website or other location that requires the stakeholder to initiate a process to check whether changes have been made. A process involving e-mails to the stakeholders to inform them of updates is considered an acceptable example of initiation.	It should be made clear to stakeholders when and to what extent changes are made to the 3DS SDK. Indicators of changes may include the advancement of version numbers or some other indicator such that any 3DS SDK stakeholders can easily associate a software release with the changes included in that release. Without change details or release notes, the impact of 3DS SDK changes—such as changes/updates and security patches—might not be fully realized and could have unintended consequences. The impact of the change should be documented so that all affected parties can plan appropriately for any processing changes.
	T.5.2.2 The tester shall examine vendor materials and other evidence to confirm that the procedures for updating the security guidance requires or includes the production of details indicating exactly what has changed in this new version. This may be in the form of release notes or a marked-up version of the document. A full trace of changes must be possible from the first release of the security guidance through the most current version.	
	T.5.2.3 The tester shall confirm that the change notice also includes any impacts to the functionality and/or security of the 3DS SDK, as applicable. This should include any new requirements that are conveyed to the application integrating the 3DS SDK—e.g., new APIs that must be called, or changes to the way in which entropy must be collected and passed to the 3DS SDK.	
	T.5.2.4 The tester shall confirm that the process for informing and distributing the security guidance also ensures the distribution of the change details.	

Requirements	Assessment Procedures	Guidance
5.3 Frequency of Updates to Stakeholder Guidance The 3DS SDK Vendor updates security guidance whenever changes warrant updates. The criteria for determining whether updates are necessary are clearly defined by the vendor and are reasonable. At a minimum, security guidance is reviewed at least annually, and updates made whenever new or changes to functionality, security features, APIs, or configurable settings are introduced.	**T.5.3.1** The tester shall examine vendor materials and other evidence to confirm that the vendor has a documented policy and procedure to identify when updates to the security guidance are necessary.	It is imperative that vendor security guidance reflect the current functionality and configuration options available in the 3DS SDK. Otherwise, new features and options might be misconfigured by the implementer. If changes to the 3DS SDK do not warrant updates to security guidance, then the justification for not making updates to implementation guidance should be reasonable. The criteria for making such determinations should also be documented to ensure consistency in application.
	T.5.3.2 The tester shall examine vendor materials and other evidence to confirm that the vendor has a documented policy and procedure requiring regular reviews of the security guidance. At a minimum, there must be a procedure to review the 3DS SDK and confirm any updates required at least annually.	
	T.5.3.3 Where possible, the tester shall obtain previous versions of the security guidance and confirm that they have been updated and published in accordance with the vendor policy and procedure.	
	T.5.3.4 The tester shall interview a sample of the personnel responsible for updating the security guidance to confirm that they understand the policy and procedures for this, as well as their responsibility for maintaining and updating this document.	
	T.5.3.5 The tester shall confirm that the vendor policy and procedures require updates to the security guidance when new or changes to functionality, security features, APIs or configurable settings are introduced.	
	T.5.3.6 The tester shall interview a sample of the personnel responsible for updating the security guidance to confirm that they understand that the security guidance must be updated when new or changes to functionality, security features, APIs or configurable settings are introduced.	