Global Platform®

Securing the digital future

GlobalPlatform Technology

CSP Guidance for Applet Developer

Version 0.0.0.11

Public Review

June 2026

Document Reference: GPC_GUI_237

THIS SPECIFICATION OR OTHER WORK PRODUCT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE COMPANY, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT.

Contents

Tables

Guidelines

1 INTRODUCTION

The Cryptographic Service Provider (CSP) is a cryptographic abstraction layer that aims to simplify security evaluations by separating cryptographic functionality from the application logic. Applications that use a CSP for their cryptographic and security-related operations can benefit from the CSP's evaluated and approved security functions, reducing the evaluation effort for the application itself. So-called Client Applications are typically Java Card applets operated on a CSP-enabled Secure Element platform utilizing CSP services.

This document defines the security guidelines that a Client Application must meet to gain the certification benefits. The guidance applies to Client Applications using a CSP compliant with *GlobalPlatform Technology Card Specification - Amendment N: Cryptographic Service Provider* ([GP Amd N]).

1.1 Audience

This document provides a set of guidelines for use by:

- Application Developers (developing Client Applications that utilize a CSP)
- CSP Admins (configuring a CSP instance on behalf of the Application Developer)
- Evaluation Laboratories (evaluating Client Applications that utilize a CSP)

1.2 IPR Disclaimer

Attention is drawn to the possibility that some of the elements of this GlobalPlatform specification or other work product may be the subject of intellectual property rights (IPR) held by GlobalPlatform members or others. For additional information regarding any such IPR that have been brought to the attention of GlobalPlatform, please visit https://globalplatform.org/specifications/ip-disclaimers/. GlobalPlatform shall not be held responsible for identifying any or all such IPR, and takes no position concerning the possible existence or the evidence, validity, or scope of any such IPR.

1.3 References

This section lists references applicable to this specification. The latest version of each reference applies unless a publication date or version is explicitly stated.

Table 1-1: Normative References

Standard / Specification	Description	Ref
GlobalPlatform Amendment N CSP GPC_SPE_230	GlobalPlatform Technology Card Specification Amendment N: Cryptographic Service Provider June 2026 Version 1.0 or higher	[GP Amd N]
GlobalPlatform CSP API	GlobalPlatform API (org.globalplatform.csp) Java Card API and Export File for CSP Version 1.0 or higher	[GP CSP API]
GlobalPlatform API ASN.1 CSP Protocol	GlobalPlatform API (csp-protocol.asn) ASN.1 Definition File for CSP Version 1.0 or higher	[GP CSP ASN1]

Standard / Specification	Description	Ref
Java Card API	Java Card™ Platform Application Programming Interface, Classic Edition https://docs.oracle.com/javacard/3.0.5/api/index.html Version 3.0.5 or higher	[JC API]

Table 1-2: Informative References

Standard / Specification	Description	Ref
GlobalPlatform Card Specification GPC_SPE_034	GlobalPlatform Technology Card Specification v2.3 or higher	[GP Card Spec]

1.4 Terminology and Definitions

Selected terms used in this document are included in Table 1-3.

Table 1-3: Terminology and Definitions

Term	Definition
Client Application	An Application, installed and operated on the same platform as the CSP Application, that utilizes cryptographic services provided by the CSP Application, which are specifically tailored to a particular use case. Each Client Application, identified by its AID, must be registered with the CSP Application to access these services. Once registered, Client Applications can invoke the CSP Services via the CSP API, using CSP Resource Identifiers that refer to the cryptographic keys and algorithms preconfigured by the CSP Admin.
CSP Admin	An entity authorized to access the Security Domain associated with the CSP Application. This entity is responsible for setting up and configuring a CSP Instance before it can be used by CSP Clients. This configuration process utilizes the GlobalPlatform Personalization interface ([GP Card Spec]) to, among other actions, create keys, certificates, and passwords as CSP Resources; configure their cryptographic algorithms; and grant usage permissions.
CSP Application (aka CSP)	An Application that provides CSP Services according to this specification. It is instantiated from the CSP ELF through the GlobalPlatform INSTALL Command, using a standardized CSP ELF AID. Multiple CSP Applications can coexist on the same Secure Element, each within its own Security Domain (SD), representing a dedicated use case. The CSP Application is configured by the CSP Admin via the CSP Protocol over the GlobalPlatform Personalization interface, customizing it for the specific use case, with the necessary cryptographic resources. Its cryptographic services can be utilized by registered Client Applications via the CSP API.
CSP Configuration (aka configuration)	A security configuration that contains all cryptographic keys, certificates, and passwords required for a specific use case, along with their purpose, algorithm, access control, and other parameters necessary for the CSP to provide use case-specific cryptographic services through the CSP API.

1.5 Abbreviations

Table 1-4: Abbreviations

Abbreviation	Meaning
CSP	Cryptographic Service Provider
PKI CRL	Public Key Infrastructure Certificate Revocation Lists
SD	Security Domain

1.6 Revision History

GlobalPlatform technical documents numbered *n*.0 are major releases. Those numbered *n*.1, *n*.2, etc., are minor releases where changes typically introduce supplementary items that do not impact backward compatibility or interoperability of the specifications. Those numbered *n.n*.1, *n.n*.2, etc., are maintenance releases that incorporate errata and clarifications; all non-trivial changes are indicated, often with revision marks.

Table 1-5: Revision History

Date	Version	Description
Jun 2025	v0.0.0.6	Committee Review
Jun 2026	v0.0.0.11	Public Review
TBD	v1.0	Public Release

2 GUIDELINES

This chapter defines general guidelines for Client Applications. Each guideline includes a rule that describes what must be done, a rationale explaining why it is necessary, and suggested inputs, such as application documentation or source code, that can be used for validating the rule. (The inputs are described in section 3.)

A Client Application that fails to meet a rule in this document presents a potential risk to the security of the product. The risk must be managed by the issuer of the product.

2.1 Application Implementation

This section lists rules that must be considered by the Application Developer when implementing Client Applications.

2.1.1 Excluded Java Card Cryptographic Operations

Use cryptographic operations provided by the CSP, not their equivalents from the Java Card API ([JC API]).

Guideline 1: Implementation – Excluded Java Card Cryptographic Operations

Rule	The Client Application shall not use any of the following security-related packages, classes, or interfaces from the Java Card Framework ([JC API]) for functionality that is subject to security evaluation: • javacard.security.KeyAgreement • javacard.security.KeyBuilder • javacard.security.KeyPair • javacard.security.Signature • javacard.security.Signature.OneShot • javacard.framework.PIN • javacard.framework.OwnerPINx • javacard.framework.OwnerPINxWithPredecrement • javacard.framework.OwnerPIN • javacard.framework.OwnerPINBuilder • javacardx.crypto.*
Rationale	The platform-independent security measures guaranteed by the CSP platform do not apply to the Java Card Framework interfaces listed in this rule. CSP platform certification does not allow further security guidance. Java Card platform allows for vendor-specific security guidance, but a Client Application is not required to fulfil vendor-specific security guidance for the Java Card platform.
Input	*Binaries of the Application* or Source Code of the Application

2.1.2 No Cryptographic Libraries

Use cryptographic operations provided by the CSP and not those from any other cryptographic library.

Guideline 2: Implementation – No Cryptographic Libraries

Rule	The Client Application shall not use cryptographic operations from other libraries not already listed in rule 2.1.1 for functionality that is subject to security evaluation, e.g., such as: • Signature and Cipher operations • PIN and Password verifications • Secure Message authentication • Attestation creation or verification
Rationale	The platform-independent security measures guaranteed by the CSP apply only to security-related functionality that exclusively relies on CSP services. Other cryptographic libraries are not considered to be included in the CSP platform certification.
Input	*List of External Libraries* and *Source Code of the Application*

2.1.3 No Cryptographic Implementations

Use cryptographic operations provided by the CSP, and do not implement any cryptographic operations within the Client Application.

Guideline 3: Implementation – No Cryptographic Implementations

Rule	The Client Application shall not implement any cryptography for functionality that is subject to security evaluation, e.g., such as: • Signature and Cipher operations • PIN and Password verifications • Secure Message authentication • Attestation creation or verification
Rationale	The platform-independent security measures guaranteed by the CSP apply only to security-related functionality that exclusively relies on CSP services. Implementing cryptography refers to realizing cryptographic functionality within application code instead of using the cryptographic services provided by the CSP.
Input	Source Code of the Application

2.1.4 Sensitive Results

Ensure that the Client Application revalidates security-related values returned by the CSP if their disclosure or modification could compromise the security of the Client Application.

For example, when computing a signature using CSP services, the CSP internally stores the length of the generated signature. When the Client Application subsequently calls `assertSensitiveResult`, passing the length previously returned by the CSP, the CSP compares this value with the internally stored reference. If a discrepancy is detected, the CSP interrupts the execution control flow of the Client Application, efficiently detecting unauthorized modifications of the signature.

Note: A list of CSP methods that support sensitive result verification is available in [GP Amd N]. These methods are also highlighted in the CSP API documentation ([GP CSP API]).

Guideline 4: Implementation – Sensitive Results

Rule	The Client Application shall invoke the `assertSensitiveResult` method from the CSP API class `org.globalplatform.csp.api.CSPService` to verify the outcome of CSP operations for all methods supporting sensitive result verification, as listed in [GP Amd N] section 5.1.3.3.
Rationale	Sensitive result verification protects the Application-to-CSP interface against control flow manipulation (e.g., glitch attacks), by ensuring that CSP return values are unmodified. It is applied as a general countermeasure, even if not required on a specific platform.
Input	*Binaries of the Application* or *Source Code of the Application*

2.1.5 Sensitive Arrays

Ensure that the Client Application protects integrity-sensitive array parameters passed to CSP services if their modification could compromise the security of the Client Application.

For example, when creating a signature, the data to be signed should be protected using Java Card Sensitive Arrays ([JC API]) before being submitted to the CSP. These arrays include an integrity control element that is verified by the platform when invoking the `assertIntegrity` operation. If an inconsistency is detected, the platform interrupts the execution control flow of the Client Application.

Note: A list of CSP methods supporting sensitive arrays is available in [GP Amd N]. Affected method parameters are also highlighted in the CSP API documentation ([GP CSP API]).

Guideline 5: Implementation – Sensitive Arrays

Rule	The Client Application shall use `javacard.framework.SensitiveArrays` ([JC API]) to safeguard all buffer parameters that require integrity protection, and shall invoke `assertIntegrity` on sensitive output buffers, as specified in [GP Amd N] section 5.1.3.4.
Rationale	Sensitive arrays protect the Application-to-CSP interface against data tampering and unauthorized access by ensuring the integrity of sensitive data transported in buffers. They are applied as a general countermeasure, even if not strictly required on a specific platform.
Input	*Binaries of the Application* or *Source Code of the Application*

2.1.6 Error Handling

Ensure that the Client Application removes temporary sensitive information (e.g., session data) from the CSP Application if a security-related error or exception occurs during CSP operation.

Guideline 6: Implementation – Error Handling

Rule	If the Client Application detects an error with potential security impact during a CSP API operation, it shall delete all CSP Application session data related to this operation using the appropriate reset functions provided by the CSP API, such as: • `org.globalplatform.CSP.api.SecureChannelService.resetSecurity` • `org.globalplatform.CSP.api.ResourceService.clear` • `org.globalplatform.CSP.api.ResourceService.clearTransient`
Rationale	Removing sensitive data from the CSP Application prevents its recovery and potential misuse.
Input	*Source Code of the Application* or both *Functional Test Case* and *Binaries of the Application*

2.1.7 Compliance with CSP Configuration

Ensure that the Client Application verifies whether the CSP Application is configured with the cryptographic keys, certificates, passwords, and services required for its operation.

The CSP Admin is responsible for creating and versioning the CSP Configuration, which includes both general CSP settings and use-case-specific keys and algorithms.

Guideline 7: Implementation – Compliance with CSP Configuration

Rule	The Client Application shall use the methods `CSP.getConfigVersion` and `CSP.getConfigName` from the CSP API class `org.globalplatform.csp.api.CSP` to verify that the CSP is configured with the correct version and use case required by the Client Application. If the CSP Configuration is incompatible, the Client Application shall abort its operation.
Rationale	The security guarantee of the CSP remains valid only if the Client Application utilizes a CSP that is configured correctly to meet the specific needs of the Client Application.
Input	*Source Code of the Application* or both *Functional Test Case* and *Binaries of the Application*

2.1.8 Security Conditions with Policies

Use the CSP's policy feature instead of implementing security-related conditional logic within the Client Application that, if disclosed, modified, or bypassed, could compromise the security of the Client Application.

For example, if encryption shall only be possible after successful PIN verification, do not implement this check in the Client Application. Instead, configure a `POLICY_PASSWORD` on the encryption key, enabling the CSP to enforce this condition internally and restrict the cipher operation to cases where the associated PIN resource has been successfully verified.

Guideline 8: Implementation – Security Conditions with Policies

Rule	The Client Application shall not implement security conditions if equivalent enforcement can be achieved using CSP policies. Equivalent enforcement refers to security conditions that provide at least the same level of protection, resistance to bypass, and assurance as enforcement through CSP policies, as required for the applicable certification or evaluation context. If CSP policies are not applicable, the evaluation of such conditions must be assessed within the assurance level of the Client Application, and remaining risks shall be mitigated by other appropriate means.
Rationale	Using CSP policies instead of application-side conditions reduces the risk of bypass or manipulation and ensures that security checks are handled within the CSP's protected environment.
Input	*Source Code of the Application*

2.1.9 Asset Management

Exclusively use the CSP to manage assets that are subject to security evaluation.

For example, the actual values of private or symmetric keys, PINs, or passwords are processed within the secure environment of the CSP. Furthermore, if the Client Application needs to transmit user credentials to an off-card system, the CSP's cipher transformation and confidential data transfer features allow secure access and transmission of the credential without reading or storing its raw value within the Client Application.

Guideline 9: Implementation – Asset Management

Rule	Assets requiring a high level of protection shall be processed exclusively within the CSP. The Client Application shall not access or use these assets directly. Furthermore, the Client Application shall not store or handle assets requiring a higher security assurance level than that for which the Client Application is evaluated. If sensitive assets need to be transferred to off-card applications, the Client Application shall use the CSP's confidential data transfer or cipher transformation feature. If it is not possible to handle a sensitive asset entirely within the CSP, the associated security requirements must be addressed within the Client Application's assurance level, and any remaining risks shall be mitigated by other appropriate means.
Rationale	The platform-independent security measures guaranteed by the CSP apply only to assets that are exclusively managed and protected by the CSP.
Input	Source Code of the Application

2.1.10 Cipher Initialization Data

Ensure that the Client Application handles cipher initialization data, such as Initialization Vectors (IVs) or nonces, in accordance with the CSP specification.

For example, reusing IVs can compromise the security of encrypted data. Furthermore, initialization data may need to be unpredictable, random, or formatted in a specific way depending on the cipher mode.

Guideline 10: Implementation – Cipher Initialization Data

Rule	The Client Application shall provide cipher initialization data (e.g., IVs or input padding) in accordance with the cipher algorithms and modes defined in [GP Amd N] section 6.1.1.5.
Rationale	Incorrect or repeated use of initialization data can weaken the security of encryption and may enable attacks such as plaintext recovery or pattern detection. Following the CSP's initialization requirements ensures that cryptographic operations remain secure.
Input	Source Code of the Application

2.1.11 Key Derivation Input Data

Ensure that the Client Application provides input data for key derivation operations according to the algorithm-specific requirements defined by the CSP.

For example, HKDF supports both a salt and optional application context information, while PBKDF2 requires a salt to prevent rainbow table attacks. The required format and usage of input data depend on the selected key derivation algorithm.

Guideline 11: Implementation – Key Derivation Input Data

Rule	The Client Application shall provide input data for key derivation operations in accordance with the requirements defined in [GP Amd N] section 6.7.1.8.
Rationale	Incorrect use or omission of derivation input data can weaken key separation, enable key recovery attacks, or violate algorithm constraints. Following the algorithm-specific rules ensures secure and interoperable key derivation within the CSP.
Input	Source Code of the Application

2.2 CSP Configuration

This section lists the rules that a CSP Admin must follow when configuring cryptographic keys, PINs, algorithms, and services of a CSP instance for a specific Client Application.

2.2.1 Versioning

Ensure that the CSP Configuration is versioned in a way that allows the Client Application to verify compatibility with its use case (see section 2.1.6, Error Handling).

Guideline 12: Configuration – Versioning

Rule	The CSP Admin shall assign a unique version and identifier to the CSP Configuration and shall update the version whenever a configuration change affects the security of the Client Application, enabling Client Applications to verify that they are using a CSP configured in accordance with their use case.
Rationale	A mismatch between the Client Application and the CSP Configuration identifier or version indicates a configuration inconsistency that may affect the security behavior of the Client Application.
Input	*CSP Configuration*

2.2.2 Client Application Registration

Ensure that Client Applications are registered to the CSP with both their own AID and the AID of their associated Security Domain.

Guideline 13: Configuration – Client Application Registration

Rule	The CSP Admin shall register each Client Application that uses the CSP Instance with both its Application Identifier (AID) and the AID of its associated Security Domain.
Rationale	If the CSP Configuration contains Client Applications that are registered only by their AID, a malicious Client Application with the same AID but installed in a different Security Domain could potentially misuse the CSP instance.
Input	*CSP Configuration*

2.2.3 Client Application Registration Measures

Ensure that Client Application integrity and authenticity are verified by configuring additional registration measures supported by the CSP where applicable to the application and use case.

Possible registration measures include verifying the Load File Data Block Hash (LFDBH) of the Client Application binaries or enforcing Data Authentication Pattern (DAP) verification.

Such checks are performed by the CSP when the Application retrieves the CSP instance via the Global Service.

Guideline 14: Configuration – Client Application Registration Measures

Rule	The CSP Admin should configure additional Client Application registration measures supported by the CSP, such as LFDBH and DAP verification.
Rationale	The LFDBH verification checks the integrity of the Client Application binary files. The DAP verification checks the authenticity of the Client Application load file.
Input	*CSP Configuration*

2.2.4 Resource Creation

Ensure that cryptographic keys, certificates, and passwords required by the Client Application are correctly configured within the CSP.

Guideline 15: Configuration – Resource Creation

Rule	The CSP Admin shall create and initialize all cryptographic keys, certificates, and passwords as required by the Client Application.
Rationale	Improper resource configuration, such as missing cryptographic values or incorrect attributes, may cause unexpected Client Application behavior and could lead to security vulnerabilities.
Input	*CSP Configuration, Application Documentation*, and *Use Case Documentation*

2.2.5 Access Control Config

Ensure that access rules for CSP resources and services are configured minimally, allowing the Client Application to invoke only those CSP services essential for its proper functioning, and limiting permissions strictly to what is necessary.

Guideline 16: Configuration – Access Control Config

Rule	The CSP Admin shall configure access control for each key, certificate, and password resource to enforce only the minimal permissions required by the Client Application. No additional permissions shall be granted.
Rationale	Granting unnecessary access rights can increase the potential for security breaches. Such excess permissions should be avoided to minimize the risk of attacks.
Input	*CSP Configuration* and *Application Documentation*

2.2.6 Policy Config

Ensure that required security conditions are enforced using CSP policy rules instead of conditional logic in the Client Application (see section 2.1.8, *Security Conditions with Policies*).

Guideline 17: Configuration – Policy Config

Rule	The CSP Admin shall define policy rules for security conditions required by the Client Application, enabling the Client Application to minimize conditional logic within its own code.
Rationale	Reducing reliance on conditional logic lowers the risk of security vulnerabilities caused by incorrect execution or exploitation through fault injection and logical attacks. Using CSP policy configurations ensures consistent and secure enforcement of such conditions.
Input	*CSP Configuration* and *Application Documentation*

2.2.7 Algorithm Config

Ensure that each cryptographic key, certificate, and password resource required by the Client Application is configured with its intended use and allowed algorithms.

For example, configure a key so that it can be used solely for the CSP key attestation service, ensuring it is not permitted for general-purpose signing.

Guideline 18: Configuration – Algorithm Config

Rule	The CSP Admin shall configure all security-relevant parameters of each key, certificate, and password resource, such as cryptographic algorithms, usage types, validity dates, and retry counters, to the minimal settings required by the Client Application's use case.
Rationale	Incorrect or insufficient resource configuration may cause unexpected behavior and could introduce security vulnerabilities.
Input	*CSP Configuration*, *Use Case Documentation*, and *Application Documentation*

2.2.8 Time Config

Ensure that the CSP's time functionality is configured according to the needs of the Client Application.

For example, if the reference time shall be updated regularly using TA certificates exchanged during EAC authentication processes, this must be explicitly configured by the CSP Admin.

Guideline 19: Configuration – Time Config

Rule	If the Client Application requires time functionality, the CSP Admin shall configure a time synchronization process that matches the security level of the time source. This configuration shall include one or more of the time synchronization settings specified in [GP Amd N] section 6.11.1.2.
Rationale	Improper time synchronization can lead to security vulnerabilities. Malicious or incorrect timestamps may compromise data integrity, enable fraud, or disrupt transaction validation.
Input	*CSP Configuration*, *Use Case Documentation*, and *Application Documentation*

2.2.9 Audit Config

If audit logging is required, ensure that the CSP's audit settings are configured to meet the needs of the Client Application.

For example, if a uniquely numbered, timestamped log trail is needed for signature creation processes, the CSP Admin must configure the required log events and fields to the corresponding signature key.

Guideline 20: Configuration – Audit Config

Rule	If the Client Application requires audit logging, the CSP Admin shall configure the CSP to activate logging for the relevant resources. This configuration may involve one or more of the following settings: • Enable time synchronization to include timestamps in the logs. • Activate usage counters if log entries require numbering. • Configure the fields to be logged in each log entry. • Configure the events that need to be logged. • Configure the audit signing key.
Rationale	Proper configuration of the CSP is critical to ensuring the integrity and authenticity of audit logs. Timestamps and usage counters prove the sequence and completeness of recorded operations, preventing data tampering, including unauthorized deletions.
Input	*CSP Configuration*, *Use Case Documentation*, and *Application Documentation*

2.2.10 Counter Capacities

Ensure that the configured counter capacity for all counters used matches the expected increment frequency and product lifetime.

Guideline 21: Configuration – Counter Capacities

Rule	The Client Application shall not exceed the minimum number of supported changes defined for the selected counter capacity in section 6.10.1.3 of [GP Amd N]. If the expected number of increments exceeds this limit, the CSP Admin shall select a higher counter capacity.
Rationale	Exceeding the defined limit may result in counter exhaustion and service interruption.
Input	*CSP Configuration*, *Use Case Documentation*, and *Application Documentation*

2.3 Application Integration

Client Applications are typically part of a larger solution involving other off-card applications and remote provisioning of the Client Application. This section lists rules that must be considered when integrating Client Applications that utilize a CSP into such scenarios.

2.3.1 API Compliance

Ensure that the Client Application is only deployed on platforms that support the required versions of the CSP API and CSP Protocol.

For example, the deployment scripts should automatically reject installation on platforms that do not meet the required versions.

Guideline 22: Integration – API Compliance

Rule	The Client Application deployment process shall include automated technical measures to ensure that the Client Application is installed only on platforms that: • Support the minimum required CSP API version. • Support the minimum required CSP Protocol version. These checks shall be performed during installation and shall abort the process if compatibility cannot be confirmed.
Rationale	The security guarantees of the CSP apply only if the Client Application operates on a CSP-enabled platform that meets the required API and protocol versions.
Input	*Deployment Scripts* and *Deployment Documentation*

2.3.2 Feature Compliance

Ensure that the Client Application verifies that the CSP platform supports all features required for secure operation.

The `CSPEnforce` command ([GP Amd N]) can be used in a script-based check to determine which features and algorithms are supported by the platform. For example, such a check can determine whether a required signature algorithm or time support is available.

Guideline 23: Integration – Feature Compliance

Rule	During Client Application deployment, the deployment procedure shall use the `CSPEnforce` command to query all features, algorithms, data types, and operation modes required by the Client Application, as specified in [GP Amd N] section 7.3.1. If the platform does not support a required feature or algorithm, the Client Application shall not be installed on that platform.
Rationale	The security guarantees of the CSP apply only if the Client Application operates on a platform that supports all features and algorithms required by the Client Application.
Input	*Source Code of the Application* or both *Functional Test Case* and *Binaries of the Application*

2.3.3 CSP Genuineness

Ensure that the platform's identity and authenticity are verified before using the CSP platform in operational mode.

For example, deployment scripts should validate the platform using CSP system attestations before proceeding with Client Application installation.

Guideline 24: Integration – CSP Genuineness

Rule	The Client Application deployment process shall verify the genuineness of the platform using one of the following system attestations, as specified in [GP Amd N] section 5.4.1.1: • CSP Platform Attestation using the GlobalPlatform CASD mechanism • CSP Config Attestation using a CSP attestation key specific to the use case
Rationale	Only platforms with proven authenticity can provide the expected security guarantees. Skipping these checks may allow the Client Application to run in an untrusted or misconfigured environment.
Input	*Deployment Scripts* and *Deployment Documentation*

2.3.4 Administration

Ensure that the CSP cannot be manipulated or misused by untrusted entities.

For example, this can be achieved by allowing only a minimal set of trusted parties to configure the CSP.

Guideline 25: Integration – Administration

Rule	The organization providing the Client Application shall ensure that: • Only dedicated trusted entities are allowed to modify the CSP Configuration. • Only dedicated trusted entities have access to the Security Domains (SDs) of both the Application and the CSP.
Rationale	The platform-independent security measures are effective only if the CSP cannot be compromised by unauthorized or untrusted entities.
Input	*Roles Documentation*

2.3.5 Key Distribution

Ensure that any secret keys or passwords imported to the CSP are handled exclusively by trusted entities.

For example, confidential data transfer keys may be shared with off-card applications to encrypt user data. In such cases, the Client Application provider must ensure that these keys are protected against manipulation or misuse.

Guideline 26: Integration – Key Distribution

Rule	All key and password values imported to the CSP shall be handled exclusively by trusted entities.
Rationale	Secrets managed by the CSP but handled outside its scope pose a security risk. Appropriate security measures must be implemented to prevent their compromise.
Input	*CSP Configuration*, *Roles Documentation*, and *Key Management Documentation*

2.3.6 Key Revocation

Ensure that expired or revoked keys, certificates, or passwords are removed from the CSP by updating the configuration.

For example, since the CSP does not support automatic revocation or Certificate Revocation Lists (CRL), compromised keys must be manually replaced or deleted by re-importing the affected values.

Guideline 27: Integration – Key Revocation

Rule	The organization providing the Client Application shall establish an organizational or scripted process to handle the revocation of keys, certificates, or passwords by updating the CSP Configuration and re-creating or re-importing the affected resource values.
Rationale	Expired or compromised cryptographic material that remains active in the CSP may lead to security vulnerabilities. These risks must be mitigated through proper revocation handling.
Input	*Key Management Documentation*

2.3.7 Time Management

Ensure that the CSP's system time is properly initialized if the Client Application relies on time-based functionalities such as validity checks.

For example, the Client Application may regularly provide an updated reference time received from an external time source, or the CSP may update its reference time using Terminal Authentication (TA) certificates to estimate a system time.

Guideline 28: Integration – Time Management

Rule	If the Client Application relies on external time sources, the process of setting the reference time to the CSP shall meet the security requirements of the Client Application.
Rationale	Improper initialization of time may disable time-based features of the CSP and could lead to security vulnerabilities, especially if the Client Application depends on these functions.
Input	*Time Management Documentation*

2.3.8 Audit Processing

Ensure that the Client Application regularly fetches audit log messages from the CSP if audit logging is required.

For example, if log messages are not fetched in time, older entries may be overwritten due to the CSP's limited storage capacity.

Guideline 29: Integration – Audit Processing

Rule	If audit functionality is required, the process for fetching log messages shall meet the security requirements of the Client Application. It shall ensure that log messages are fetched in time and that the retrieved logs can be processed as required by the use case.
Rationale	If log retrieval is delayed, the CSP may overwrite old entries, discard logs, or even switch to a restricted mode (e.g., AUDIT_MODE_STRICT). These behaviors can result in loss of audit data or limited CSP functionality, which may affect the security of the Client Application.
Input	*Audit Logging Documentation*

3 INPUT

To demonstrate that a Client Application adheres to the rules outlined in this document, the input documents described in this section must be made available to the evaluator upon request.

3.1 Source Code of the Application

The source code of the Client Application must be made accessible to the evaluation body for assessment. If necessary, the evaluator may need to physically inspect it on-site.

Specifically, rules that verify the absence of certain elements, such as the Client Application not implementing its own cryptography, require manual code inspection.

3.2 Binaries of the Application

Tools may assist in the evaluation process. Rules that require repetitive, definitive checks can be efficiently performed by automatic verification tools on the CAP files of the Client Application.

3.3 Deployment Scripts

Client Applications installed on a previously unknown platform type must include checks to ensure that the platform complies with their requirements. Therefore, the deployment scripts must be made accessible to the evaluator for assessment.

3.4 Functional Test Case

Some rules can be assessed through functional test cases.

3.5 CSP Configuration

The CSP Configuration may either be provided as part of the installation script used to create and initialize the CSP Instance, or provided as a byte array in the ASN.1 format specified in [GP CSP ASN1]. Supporting tools could assist here by converting this byte array into a human-readable format.

3.6 Documentation

The guidelines in this document necessitate the clarification of various aspects of the Client Application and its integration into the technical and organizational environment.

3.6.1 Application Documentation

The application documentation should include:

- **Overview:** Describes the architecture of the Client Application, including its main interfaces and the underlying technology stack.
- **Asset Management:** Details which assets are managed within the CSP and which are not, along with the reasons why certain assets are handled outside the CSP.
- **Version Information:** Specifies the versions of the Client Application, CSP API, and CSP Protocol used.

3.6.2 Use Case Documentation

The use case documentation should include:

- **Purpose:** Explains the purpose of the Client Application, including its high-level functions.
- **Application's Role:** Explains how the Client Application fits into the broader solution, including its contributions to the overall system.
- **Application's Integration:** Describes the Client Application's expected interactions with end-users or other systems.
- **Workflows:** Lists typical workflows, including conditional or alternative paths.

3.6.3 List of External Libraries

The list of external libraries includes all libraries imported and/or utilized by the Client Application. For each library, the list shall provide a detailed breakdown that highlights all external operations used to implement cryptographic or security-related functionality.

3.6.4 Deployment Documentation

The deployment documentation should include:

- **Deployment Procedure:** Describes the overall deployment process, including how to gain initial access to the platform, create Security Domains, and manage keys for these domains. It also covers the installation and activation of the CSP, along with the installation of the Client Application.
- **Target Platforms**: Defines the scope of the target platforms as broadly as possible.
- **Compatibility Checks:** Explains how the installation process verifies that the Client Application is only installed on CSP-enabled platforms whose version and variant are compatible with the Application.

3.6.5 Roles Documentation

The roles documentation should include:

- **Overview:** Describes the roles and organizational structure of entities involved in the Client Application lifecycle.
- **SE Admins:** Lists the roles with administrative privileges to the platform and describes measures taken to ensure that they cannot access assets managed by the CSP Instance.
- **CSP Admins:** Lists the roles with administrative privileges to the Security Domain of the CSP Instance.
- **Application Admins:** Lists the roles with administrative privileges to the Security Domain of the Application.
- **Remote Entities:** Lists roles for remote entities that have access to assets imported to the CSP.

3.6.6 Key Management Documentation

The key management and revocation documentation should include:

- **Imported CSP Resources:** Lists all resources that are imported to the CSP, including descriptions of how they are created, who creates them, to which entities they are distributed, and how measures are taken to prevent their compromise.
- **Revocation:** Specifies the procedures to follow when cryptographic keys or other sensitive assets managed by the CSP are expired or compromised. This includes steps for revocation, notifying stakeholders, and mitigating risks to ensure that the security of the Client Application remains intact even in adverse scenarios.
- **Usage Limits Control:** Describes how the usage limits for cipher operations ([GP Amd N] section 6.1.1.3) and key derivation operations ([GP Amd N] section 6.7.1.6) are respected.

3.6.7 Time Management Documentation

The time management documentation should include:

- **Time Synchronization:** Describes the procedures for updating the CSP's reference time.
- **Authentic Time Source:** Explains the measures applied to guarantee the authenticity of the time source, accompanied by an explanation of why the method is deemed adequate.

3.6.8 Audit Logging Documentation

The audit logging documentation should include:

- **Log Format:** Lists the CSP Configuration details that ensure the integrity and authenticity of generated log files, such as the utilization of counters, timestamps, and data identifiers within log entries.
- **Log Fetching:** Describes the procedures for fetching the log messages.
- **Log Evaluation:** Explains the process for evaluating the log messages.