Cryptographic Services Provider (CSP)

Zertifizierung von sicherheitskritischen Anwendungen basierend auf Secure Elements

Annegret Schöffel, BSI, Referat D13

Federal Office
for Information Security

Januar 2026

Gegenmaßnahmen für Chip-Angriffe

Zertifizierung der App/Applet erzwingt konkrete Gegenmaßnahmen

Ausgangslage:

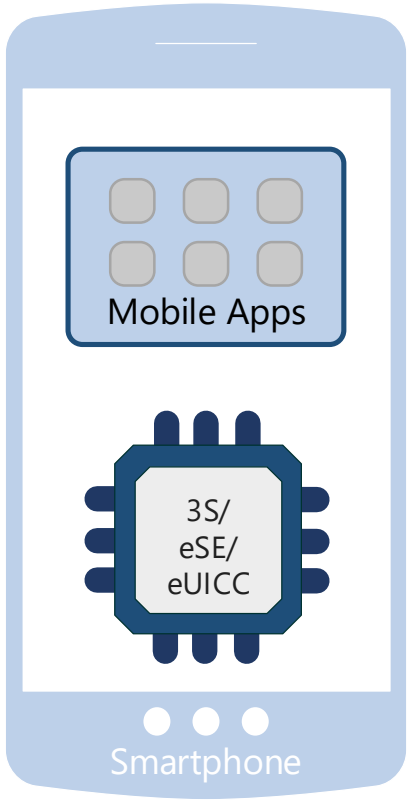
- Mobile Geräte (z. B. Smartphones) enthalten bereits hoch zertifizierte Sicherheitschips
- Common Criteria, hohes Vertrauensniveau – z. B. VAN.5

Warum bleibt ein Risiko?

- Sicherheitschips sind nur dann sicher, wenn sie korrekt genutzt werden
- Falsche Nutzung durch: Apps, Applets oder Schnittstellenlogik kann neue Angriffsflächen eröffnen

Konsequenz

- Sicherheit kann nicht allein durch den Chip garantiert werden
- Zertifizierung der Nutzung (Applet / App) notwendig
- Zertifzierung macht Angriffsflächen sichtbar und erzwingt konkrete Gegenmaßnahmen



eSE, eSIM, eUICC - bereits heute **hoch zertifizierte Sicherheitsbausteine** in modernen Smartphones

Problem

Zerifizierbarkeit für hohes Vertrauensniveau skaliert schlecht

Anwendungen auf mobilen Endgeräten werden in der Regel:

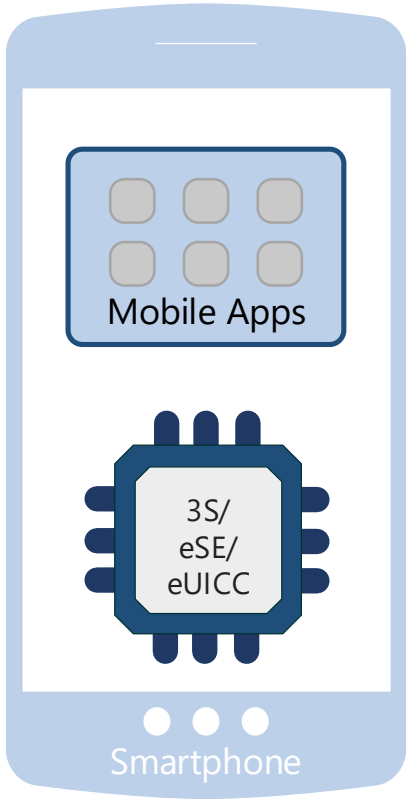
- **erst nach Auslieferung an den Endkunden installiert und eingerichtet**
- **auf einer großen Vielfalt unterschiedlicher Geräteplattformen betrieben.**

Zertifizierung nach klassischem Ansatz

- Nach klassischer Common-Criteria-Composite-Evaluierung
- muss die Anwendung für jede mögliche Hardwarekonfiguration neu bewertet werden

→ **Ergebnis: Zertifizierungsaufwand steigt nahezu exponentiell.**

Classical Smartcard



Vollständig konfiguriert vor Auslieferung, z.B.:

- SIM Karte
- Bankkarte
- Ausweiskarte

Konfiguration erst beim Endkunden (Over the Air), z.B.:

- eSIM
- EUDI-Wallet

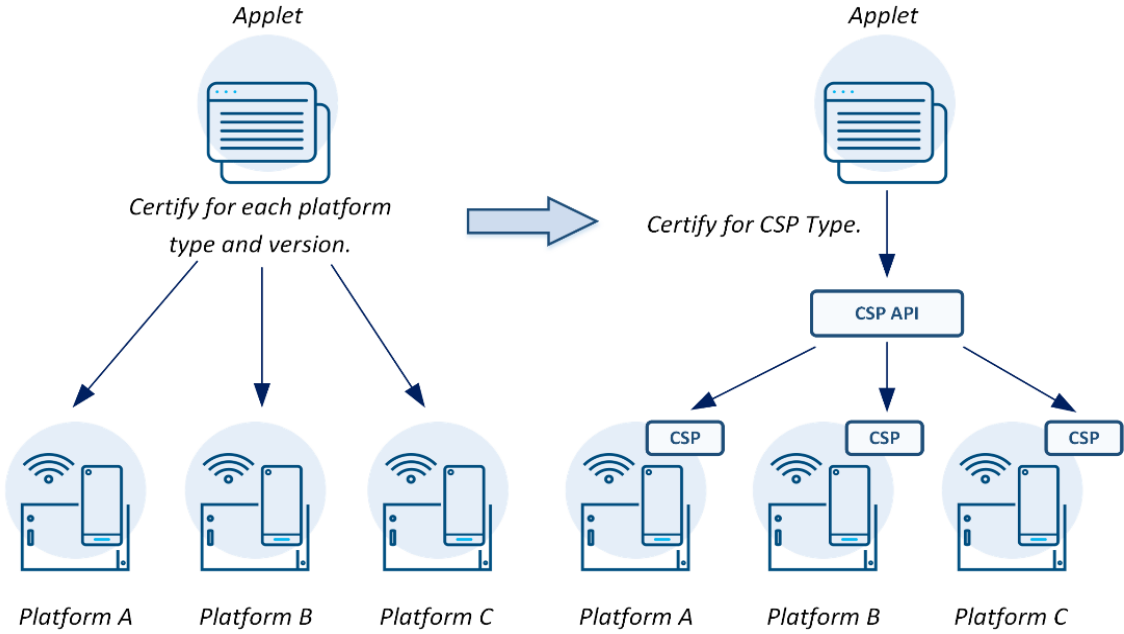
Ziel des Cryptographic Service Providers (CSP)

Eliminate the need for individual physical testing on each platform for Applications that employ the CSP API for all their security-related functions.

1) Der CSP selbst wird für jede Plattform einzeln nach dem traditionellen Composite-Ansatz zertifiziert.
2) Die Anwendung wird einmalig unter Verwendung einer repräsentativen CSP-Plattform zertifiziert.
3) Für Plattformen, die zur CSP-API kompatibel sind, ist keine erneute Zertifizierung der Anwendung erforderlich.

Die Methodik

- beseitigt das Skalierungsproblem herkömmlicher Composite-Evaluierungen und
- liefert vergleichbare und aussagekräftige Sicherheitsaussagen für CSP-fähige Plattformen.



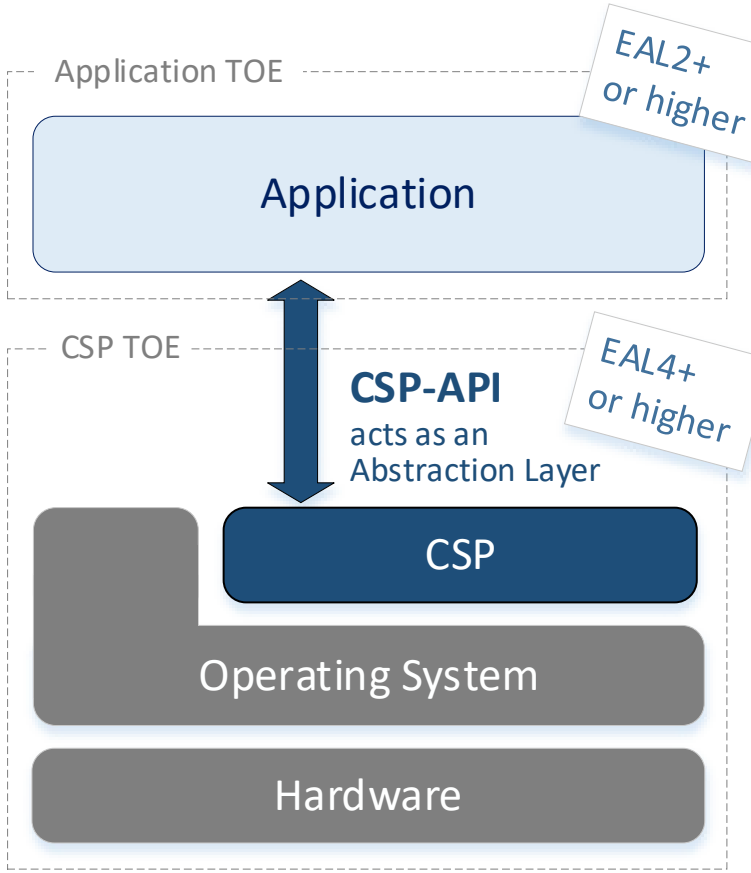
→ **CSP-based Evaluierung:** Ein **hybrider Ansatz** zwischen klassischer Softwareevaluierung und der Composite-Zertifizierung

Wie funktioniert der CSP?

Charakteristik der CSP-API

Die CSP-API ist eine **harmonisierte Schnittstelle**, die alle sicherheitsrelevanten Funktionen abdeckt, die für typische Smartcard-Anwendungen erforderlich sind.

1) Die Implementierung der CSP-API wird als **Teil des CSP-TOE evaluiert**.
2) Der CSP wird mindestens nach **EAL4 AVA_VAN.5**, in einer Composite-Evaluierung (einschließlich Betriebssystem und Hardware) zertifiziert.
3) Der CSP speichert und verarbeitet seine Konfiguration, Assets, sensiblen Daten sowie die erforderlichen Managementfunktionen in einer **manipulationsgeschützten Umgebung**, die für Anwendungen nicht direkt zugänglich ist und nicht exponiert werden kann.
4) Die CSP-basierte Evaluierungsmethodik stellt sicher, dass Anwendungen **ausschließlich die vom CSP bereitgestellten Sicherheitsleistungen** nutzten, und keine eigenen Sicherheitsmechanismen außerhalb der Vorgaben der der generischen CSP-Guidance implementieren.



Wie funktioniert der CSP?

Beispiel zur Nutzung der CSP-API

Anwendung auf Basis Java-Card-API:

```
keyPair = new KeyPair(keyType, keySize);
keyPair.genKeyPair();
pin = new OwnerPin(tryLimit, maxSize);

sig = Signature.getInstance(algorithm);
sig.init(keyPair, signMode);

if (pin.check(inputPin)) {
  output = sig.sign(inputData);
}
```

Anwendungen auf Basis CSP-API:

```
pinService = csp.getPasswordService();
sigService = csp.makeSignatureService();
sigService.init(keyId, signMode);
..
pinService.check(pinId, inputPinData);
try {
  output = sigService.sign(inputData);
} catch (CSPException e) {
  // handle PIN is not in state authenticated
}
```

Wie funktioniert der CSP?

Managementfunktionen des CSPs

- **CSP-Konfiguration**: Bevor Anwendungen den CSP nutzen können, muss dieser mit kryptographischem Material wie Schlüsseln, Zertifikaten, Passwörtern und weiteren Daten initialisiert werden.
- **Key Management:** Der CSP verwaltet Schlüsselparameter (z. B. Key-Type, Key-Size), Algorithmen und Nutzungsbeschränkungen (z. B. ein ausschließlich zur Verschlüsselung zugelassener Schlüssel, soll nur mit dem AES-CBC-Algorithmus verwendet werden dürfen).
- **Access Control Rules (ACRs):** Feingranulare Zugriffsregeln pro Asset und pro Anwendung ermöglichen, dass CSP-Dienste ausschließlich von autorisierten Instanzen genutzt werden dürfen.
- **CSP Multi Application Support**: Ein CSP kann mehrfach instanziert werden; jede Instanz besitzt eine eigene Konfiguration und eigene Assets, die vollständig voneinander getrennt und isoliert sind.
- **Policies:** Dynamische Richtlinien, die innerhalb des CSP durchgesetzt werden, können den Zugriff zusätzlich einschränken. Beispielsweise kann die Effective-Access-Rules-Policy die Entschlüsselung bestimmter Datenfelder abhängig von den Access-Flags in TA Zertifikaten erlauben oder verweigern.

Wie funktioniert der CSP?

Kryptografische Funktionen des CSPs

- **Cipher and Signatures**: Fest konfigurierte kryptographische Operationen für Verschlüsselung, Entschlüsselung, Signaturerzeugung und -verifikation unter Verwendung der vom CSP verwalteten Schlüssel und festgelegten Algorithmen.
- **Encryption Transformations**: Eine atomare kryptographische Operation, die verschlüsselte Daten von einem Schlüssel/Algorithmus in einen anderen überführt.
- **Secure Messaging**: Vom CSP verwaltete gesicherte Kommunikationsprotokolle, die die korrekte Abfolge aller Schritte eines Authentifizierungsablaufs sicherstellen und dabei die vom CSP verwalteten Schlüssel und Zertifikate nutzten (z.B. EACv1 für MRTD, EACv2 für eID, PACE-only, PACE-CAM und SCP03).
- **Attestations**: Der CSP ist in der Lage, signierte Nachweise zu erstellen, die die Authentizität und Identität kritischer Komponenten bestätigen, z.B. Plattform-Attestation oder Key-Attestation.
- **Timer, Counter and Limits**: Unterstützung für Validity Dates, Timeouts und sichere Counter - vom CSP verwaltet und durchgesetzt (z.B. wird ein Schlüssel gesperrt, wenn ein Zähler sein Limit überschreitet).

Aktivitäten bei GlobalPlatform und ENISA

Internationale Aktivitäten durch das BSI zum Thema CSP

GP SSWG:

GlobalPlatform (GP) SE Specification Working Group (SSWG) entwickelt derzeit in Zusammenarbeit mit dem BSI eine detaillierte CSP-Spezifikation:

- Die Veröffentlichung der CSP-Spezifikation „Amendment N – CSP“ ist für **Q1 2026** geplant.
- Ein neues **CSP Protection Profile** befinden sich in Erstellung

ENISA EsEm:

In der ENISA ECCG Subgroup on EUCC Maintenance and Review (EsEm) wird unter Federführung des BSI eine neue Evaluationsmethodik CSP-based Evaluation diskutiert:

- Der Draft 2.0 der Methodology befindet sich derzeit in der **zweiten** Kommentierungsrunde bei der EsEm und wurde inzwischen auch zur Durchsicht an die ISAC-ISCI übermittelt.

Details der CSP-based Evaluation Methodology

Anforderungen an CSP-Implementierungen

Die CSP-basierte Evalierungsmethodology setzt folgende Dokumente voraus:

1) **CSP PP:** Ein Schutzprofil, das die Sicherheitsleistungen des CSP definiert.
2) **Harmonized CSP API:** Eine abgestimmte API, die anwendungsspezifische Sicherheitsfunktionen bereitstellt.
3) **Generic CSP Guidance:** Ein generisches Regelwerk für Anwendungen zum plattformunabhängigen Betrieb auf kompatiblen CSP-Plattformen.
4) **CSP test specification:** Eine Testspezifikation für CSP-Implementierungen zur Sicherstellung der Interoperabilität und Kompatibilität auf verschiedenen Plattformen.

Auszug aus den Anforderungen des Draft 2.0 zur CSP-based Evaluationmethodology:

REQ-CSP-PP-02: The CSP PP shall require that the CSP API implementation is included in the scope of the CSP TOE.

REQ-CSP-PP-03: The CSP PP shall require strict conformance of any ST claiming conformance to this CSP PP.

REQ-CSP-API-03: The CSP API shall use handles for asset interaction and enforce access control, preventing direct exposure of sensitive assets to Applications.

Details der CSP-based Evaluation Methodology

Zertifikat und Certification Report

Das Zertifikat muss folgende Informationen enthalten:

- **Product Name:** <<Application Name>> using <<CSP Type Name>>
- **Composition Statement:** Relies on CSP security functionality as defined by <<CSP PP>>
- **CSP API:** <<CSP API Name + Version>>
- **Evaluation Assurance Level:** <<TOE EAL>> with composition based on <<CSP PP EAL>>
- **Composition Assurance:** <<Composition Assurance Package + Components>>

Der Zertifizierungsbericht muss folgende zusätzlichen Informationen dokumentieren:

- **Representative CSP Platform used for testing:** Specification of the exact CSP platform used for testing activities during the evaluation, including details such as platform name, platform version, operating system version, CSP implementation version and tested configurations.
- **Information** that the application can be transferred to compatible CSP platforms.

Federal Office
for Information Security

Danke für ihre Aufmerksamkeit!

Annegret Schöffel

Referat-D13@bsi.bund.de

Bundesamt für Sicherheit in der Informationstechnik (BSI)

Godesberger Allee 87

53175 Bonn

www.bsi.bund.de

Follow us:

Restrisiko und Gegenmaßnahmen

Wie der CSP das Risiko hardwarespezifischer Angriffe verringert

Die CSP-based Evaluation Methodik kann **nicht** vollständig garantieren, dass eine Anwendung gegen Angriffe immun ist, die aus hardwarespezifischen Verhaltensunterschieden entstehen. Sie biete jedoch

Mechanismen zur Risikoreduktion:

- Die Angriffsfläche der Anwendung wird reduziert (z.B. keine direkte Verwendung von Jump-Instruktionen beim Aufruf von CSP-Authentisierungsfunktionen).
- Hochkritische Assets (z. B. langfristige Schlüssel) werden ausschließlich vom CSP verarbeitet.
- Hochkritische Prozesse werden ausschließlich innerhalb des CSP implementiert.
- Eine hardware-unabhängige Developer-Guidance erzwingt sicherheitsrelevante Maßnahmen (z.B. die Nutzung der sensitive results-Instruktion, die ein CSP-Ergebnis zweimal anfordert und validiert).

→ Diese Mechanismen reduzieren das Risiko erfolgreicher Angriffe und ermöglichen es dem Risiko-verantwortlichen, das verbleibende Restrisiko zu akzeptieren.