Bundesamt
für Sicherheit in der
Informationstechnik

Deutschland
Digital•Sicher•BSI•

Technische Richtlinie TR-03171

Optisch verifizierbarer kryptographischer Schutz von Verwaltungsdokumenten (Digitale Siegel)

Version 0.8

Änderungshistorie

Tabelle 1: Versionierung

Version	*Datum*	*Name*	*Beschreibung*
0.7	25.05.2021	BSI	Erstveröffentlichung
0.8	21.05.2024	BSI	Ergänzung des Statusservers; Anpassung der Profil- und Zertifikatsreferenz mittels UUID; Aktualisierung der Datentypen

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582-0
E-Mail: digitale-siegel@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhalt

1 Einleitung

Zahlreiche Verwaltungsprozesse in Deutschland resultieren in der Erstellung von Bescheiden, Zertifikaten oder sonstigen Dokumenten. Damit solche Dokumente als Nachweis gegenüber Dritten verwendet werden können, wurden sie traditionell mit Dienstsiegeln und/oder Unterschriften der Bearbeitenden versehen. Die handschriftliche Unterschrift kann in vielen Bereichen schon seit langem durch eine qualifizierte elektronische Signatur ersetzt werden, und die mit der eIDAS-Verordnung (eIDAS-VO) EU-weit eingeführten qualifizierten Siegel, die einen elektronischen Herkunftsnachweis darstellen, verfügen über dieselben mathematischen Eigenschaften.

Mit fortgeschrittenen oder qualifizierten elektronischen Signaturen oder Siegeln lässt sich die Authentizität und Integrität elektronischer Dokumente jederzeit überprüfen, solange sie in der ursprünglichen Form elektronisch vorliegen. Sofern die Dokumente jedoch nur in Papierform vorliegen oder z. B. auf einem mobilen Endgerät präsentiert werden, sind diese Integritätssicherungen nicht mehr prüfbar. Hierfür wurden optisch verifizierbare digitale Siegel entwickelt, die die wesentlichen Daten eines Nachweises in strukturierter Form und kryptographisch gesichert enthalten. Durch das Scannen eines digitalen Siegels (in Form eines Barcodes) wird ein elektronisches Dokument erzeugt, in das die mit einer Integritätssicherung versehenen Daten eingebunden sind. Sofern bei der Erstellung ein entsprechendes Zertifikat verwendet wurde, handelt es sich bei dem geschützten Datensatz um ein mit einer qualifizierten elektronischen Signatur bzw. einem qualifizierten elektronischen Siegel versehenes Dokument.

Auf diese Weise lässt sich die Authentizität und Integrität eines Verwaltungsdokuments feststellen, etwa im Rahmen der Prüfung einer Genehmigung durch Mitarbeitende einer Ordnungsbehörde, ohne dass dazu ein Zugriff auf in einem Hintergrundsystem gespeicherte Informationen zum Vorgang notwendig wäre.

Das vorliegende Dokument beschreibt einen auf (TR-03137-1) basierenden Standard zur Erzeugung digitaler Siegel auf Verwaltungsdokumenten.

2 Organisatorische Grundlagen

Sofern eine Stelle lediglich Nachweise prüfen möchte, die sie selbst erstellt und signiert hat, kann sie die dafür erforderlichen Formate und Konventionen selbst festlegen. Soll es jedoch ermöglicht werden, dass ein Verwaltungsdokument von unterschiedlichen Stellen (z. B. anderen Kommunen) validiert wird, so ist die Einhaltung einheitlicher Konventionen erforderlich. Zudem benötigt die prüfende Stelle Zugriff auf die Dokumentformate (Profile im Sinne der (TR-03137-1)) sowie auf die zur Erstellung der elektronischen Signatur / des elektronischen Siegels verwendeten Zertifikate samt zugehöriger Zertifikatskette und Rückrufinformationen.

2.1 Profilverwaltung

Damit eine Anwendung die Inhalte aller Verwaltungsdokumente strukturiert anzeigen kann, die von einer öffentlichen Stelle ausgegeben und mit einem optisch verifizierbaren digitalen Siegel versehen wurden, ist es notwendig, dass sie Zugriff auf alle zugrundeliegenden Dokumentenprofile hat. Hierzu ist eine zentrale Profilverwaltung der öffentlichen Verwaltung einzurichten, die von jeder öffentlichen Stelle Profile entgegennimmt und zusammen mit jeweils einer eindeutigen Kennung in Form einer Dokumentenprofilnummer auf einem öffentlich zugänglichen Server zur Verfügung stellt. Die Profilverwaltung muss sicherstellen, dass Profile mit einer bestehenden Dokumentenprofilnummer nicht angenommen werden. Außerdem muss ein geeigneter Authentifizierungsmechanismus implementiert werden, damit die Profilverwaltung ausschließlich Profile von autorisierten und berechtigten Stellen akzeptiert. Die benötigten Profile können dann bei Bedarf online abgerufen oder durch entsprechende Anwendungen bereits im Vorfeld für Offline-Prüfungen heruntergeladen werden.

2.2 Zertifikatsverwaltung

Während fortgeschrittene und qualifizierte Signaturen und Siegel meistens das Zertifikat enthalten, mit dem sie erstellt wurden, enthalten optisch verifizierbare digitale Siegel aufgrund der begrenzten Speicherkapazität nur eine Referenz auf ein Zertifikat, das an anderer Stelle vorliegt. Damit eine Anwendung die Integrität aller Verwaltungsdokumente prüfen kann, die von einer öffentlichen Stelle ausgegeben und mit einem digitalen Siegel versehen wurden, ist es notwendig, dass sie Zugriff auf alle verwendeten Zertifikate hat. Hierzu ist eine zentrale Zertifikatsverwaltung der öffentlichen Verwaltung einzurichten, die von jeder öffentlichen Stelle Zertifikate entgegennimmt und zusammen mit jeweils einer eindeutigen Kennung bestehend aus *Signer Identifier* („DEZV“) und *Certificate Reference* (<Zertifikatsnummer>) auf einem öffentlich zugänglichen Server zur Verfügung stellt. Dabei lautet der Signer Identifier stets „DEZV". Die Certificate Reference besteht aus einer 128-Bit langen UUID (Universally Unique Identifier) der Version 4 gemäß (RFC4122), welche mit echten Zufalls- oder Pseudo-Zufallszahlen erstellt wird. Die Zertifikatsverwaltung generiert diese UUID und sendet sie nach dem Hochladen des Zertifikats zurück. Für die C40-Kodierung der UUID werden die hexadezimalen Zeichen ohne Bindestriche und in Großbuchstaben umgewandelt.

Die benötigten Zertifikate und Zertifikatsketten können dann bei Bedarf online geprüft oder durch entsprechende Anwendungen bereits im Vorfeld für Offline-Prüfungen heruntergeladen werden. Für die Offline-Nutzung ist es notwendig, dass das Zertifikat zusammen mit der dazugehörigen *Certificate Reference* gespeichert wird. Dabei ist zu beachten, dass Zertifikate bei Offline-Verwendung gerade nicht in Echtzeit auf Gültigkeit bzw. Rückruf geprüft werden können und eine solche Prüfung daher je nach Anwendungsszenario mit den für eine Offline-Prüfung heruntergeladenen Zertifikaten regelmäßig online durchzuführen ist.

Es muss ein geeigneter Authentifizierungsmechanismus implementiert werden, damit die Zertifikatsverwaltung ausschließlich Zertifikate von autorisierten und berechtigten Stellen akzeptiert und verarbeitet.

3 Format digitaler Siegel auf Verwaltungsdokumenten

Das Format digitaler Siegel auf Verwaltungsdokument folgt grundsätzlich den Vorgaben der (TR-03137-1). Zulässige Ausprägungen sowie notwendige Abweichungen werden im Folgenden beschrieben.

3.1 Header

Die vorliegende Technische Richtlinie unterstützt ausschließlich Header der Version 4 gemäß (TR-03137-1). Die Header sind gemäß (TR-03137-1) zu befüllen; ausgewählte Header werden im Folgenden erläutert und vorgeschriebene Belegungen in Anführungszeichen angegeben.

Als Dokumentenkategorie wird die Nummer `200` für Verwaltungsdokumente vergeben. Aufgrund der zu erwartenden Vielzahl an zu siegelnden Verwaltungsdokumenten wird die jeweilige Ausprägung des Verwaltungsdokuments nicht im für Ausprägungen vorgesehenen Header-Feld *Document Feature Definition Reference*, der nur 256 verschiedene Werte zulässt, sondern in einem eigenen Feld als Dokumentenprofilnummer in der Message Zone (s. Abschnitt 3.2) codiert.

Tabelle 2: Header gemäß TR-03137-1

Position gem. (TR-03137-1)	*Inhalt*	*Wert*
`0x01`	*Version*	„`0x03`“
`0x02`	*Issuing Country*	„`D<<`“
`0x04`	*Signer Identifier and Certificate Reference*	von der Zertifikatsverwaltung ausgegebene Kennung
`0x0A+v`	*Document Feature Definition Reference*	„`01`“ für Konformität mit Abschnitt 3.2 (die eigentliche Feature Definition der Inhaltsdaten wird im ersten Feld der Message Zone gesetzt)
`0x0B+v`	*Document Type Category*	„`200`“ (Verwaltungsdokument)
`0x0A / 0x04+v`	*Document Issue Date*	Ausgabedatum des Dokuments
`0x0D / 0x07+v`	*Signature Creation Date*	Erstellungsdatum der Signatur

3.2 Message Zone

Die Message Zone ist wie in (TR-03137-1) definiert zu befüllen. Dabei ist unter Tag `0x00` die Dokumentenprofilnummer zu hinterlegen, und zwar in der folgenden Form:

Tabelle 3: Angabe der Dokumentenprofilnummer in der Message Zone

Eigenschaft	*Wert*
tag	„`0x00`“
length	Die Länge des nachfolgenden Werts, in diesem Fall einer UUID, beträgt 16 Bytes (128 Bit) und wird hexadezimal als '10' dargestellt.
value	Dokumentenprofilnummer, die als 128 Bit lange UUID im ASN.1 OCTET STRING-Format kodiert wird

Unter Tag `0x01` sind optional zwei Datumsangaben zu hinterlegen, zum einen „`validFrom`“ welche angibt ab wann das Dokument gültig ist und „`validTo`“ welches angibt bis wann das Dokument gültig ist.

Tabelle 4: Angabe der optionalen Datumsangaben „`validFrom`“ und „`validTo`“ in der Message Zone

Eigenschaft	*Wert*
tag	„`0x01`“
length	Die Länge des nachfolgenden Werts, repräsentiert die kombinierte Länge der Datumsangaben „`validFrom`“ und „`validTo`“. Jedes Datum hat eine feste Länge von 8 Bytes (64 Bit) im ASN.1 DATE-Format YYYYMMDD. Sind beide Daten vorhanden, beträgt die Länge 17 Bytes (136 Bit) inklusive eines Null-Bytes (0x00) als Trennzeichen. Ist nur ein Datum vorhanden, beträgt die Länge 9 Bytes (72 Bit), inklusive des Null-Bytes am Anfang oder Ende. Sind keine Daten vorhanden, ist die Länge 1 Byte (8 Bit), das nur das Null-Byte beinhaltet.
value	Eine Zeichenfolge, die die optionalen Datumsangaben „`validFrom`“ und „`validTo`“ enthält. Die Datumsangaben sind im Format YYYYMMDD kodiert. „`validFrom`“ wird vor dem Null-Byte platziert und „`validTo`“ folgt dem Null-Byte. Wenn nur „`validFrom`“ vorhanden ist, wird das Null-Byte direkt nach dem Datum eingefügt. Ist nur „`validTo`“ vorhanden, beginnt die Zeichenfolge mit einem Null-Byte gefolgt vom Datum.

Die Tags `0x02` bis `0x03` werden für eine spätere Verwendung reserviert. Für Inhaltsdaten stehen die Tags `0x04` bis `0xfe` zur Verfügung.

3.3 Verwendete Zertifikate

Für die Erstellung fortgeschrittener oder qualifizierter Signaturen oder Siegel können verschiedenartige Zertifikate zum Einsatz kommen:

- Fortgeschrittene Signaturen und Siegel können beispielsweise mit Zertifikaten aus der Verwaltungs-PKI erzeugt werden.
- Zur Erstellung qualifizierter Signaturen und Siegel werden qualifizierte Zertifikate eines (kommerziellen) qualifizierten Vertrauensdiensteanbieters benötigt.

Um keine zusätzlichen Einschränkungen an die zu verwendenden Zertifikate zu machen, gelten im Anwendungsbereich der vorliegenden Technischen Richtlinie die Vorgaben aus Abschnitt 2.2.1 des (DOC9303-13) nicht. Bei Verifizierung eines digitalen Siegels wird das benötigte Zertifikate nicht über die darin enthaltenen Zertifikatsdetails, sondern ausschließlich über die von der Zertifikatsverwaltung vergebene und im digitalen Siegel codierte eindeutige Zertifikatsnummer identifiziert.

4 Dokumentenprofile

Zur korrekten Codierung und Darstellung der Daten im digitalen Siegel muss ein Profil folgende Daten enthalten:

- Dokumentenprofilnummer („*profileNumber*“)

  Zur eindeutigen Identifizierung jedes Profils wird eine 128-Bit lange UUID (Universally Unique Identifier) der Version 4 verwendet, die gemäß (RFC4122) mit echten Zufalls- oder Pseudo-Zufallszahlen erstellt wird. Diese UUID ermöglicht eine dezentrale und einzigartige Kennzeichnung jedes Profils und kann von der erstellenden Stelle direkt generiert werden, ohne dass ein zentraler Erstellungsprozess notwendig ist. Nach der Erzeugung dient die UUID fortan als dauerhafte und unverwechselbare Identifikationsnummer für das jeweilige Profil. Sollte es eine Änderung an einem Profil geben, so bekommt dieses eine neue Dokumentenprofilnummer. Für die Speicherung und Übertragung der 128 Bit langen UUID muss der ASN.1 „OCTET STRING-Typ“ (Tag: `0x04`) genutzt werden. In textueller Repräsentation kann hingegen auf die Formatierung mit Bindestrichen in fünf Gruppen zurückgegriffen werden, zur Erhöhung der Lesbarkeit.

- Profilname („*profileName*“)

  Der Profilname ist ein frei wählbarer, beschreibender Name, der das Profil für Menschen leicht identifizierbar macht. Er dient der schnellen Orientierung und Unterscheidung in der Praxis und sollte daher klar und aussagekräftig sein.

- Ersteller des Profils, zuständige Stelle („*creator*“)

  Bezeichnet die Organisationseinheit oder Behörde, die für die Verwendung und Verwaltung des Profils zuständig ist. Dies umfasst die Verantwortung für die Pflege, Aktualisierung und Anwendung des Profils im Rahmen ihrer spezifischen administrativen und operativen Tätigkeiten. Die Angabe dieser Stelle stellt sicher, dass für jedes Profil klar ist, welche Einrichtung als Ansprechpartner für Rückfragen oder für die Klärung von Sachverhalten bezüglich des Profils dient.

- Kategorie des Profils („*category*“) [optional]

  Dieses Feld erlaubt eine freie Texteingabe zur weiteren Klassifizierung des Profils. Es kann zur Gruppierung oder thematischen Einordnung des Profils innerhalb der Verwaltung genutzt werden.

- Schlüssel der relevanten Verwaltungsleistung, LeiKa-ID[1] („*leikaID*“) [optional]

  Dieser Schlüssel dient zur Zuordnung des Profils zu spezifischen Verwaltungsleistungen. Wenn genutzt, werden die LeiKa-IDs als Identifikatoren für die jeweiligen Verwaltungsleistungen verwendet. Es können mehrere LeiKa-IDs angegeben werden, um das Profil mit verschiedenen Verwaltungsleistungen zu verknüpfen. Zur Trennung der LeiKa-IDs wird ein Semikolon (;) als standardisiertes Trennzeichen eingesetzt.

- Statuskennzeichen („*statusIndicator*“) [optional]

  Dieses optionale Feld zeigt den Bedarf und die Art der Anfrage an den Statusserver an. Je nach festgelegtem Wert ergeben sich unterschiedliche Anforderungen für die Überprüfung des Siegels. Folgende Optionen stehen zur Verfügung:

  - `NONE`: Keine Statusserverabfrage erforderlich. Dieser Wert wird verwendet, wenn keine Überprüfung des Siegels am Statusserver notwendig ist. (Default)

    - `BLOCKLIST`: Die Anwendung muss eine Anfrage an den Statusserver stellen, um zu überprüfen, ob das Siegel als zurückgezogen („`REVOKED`“) gelistet ist. Ist das der Fall, wird das Siegel als ungültig markiert.
    - `ALLOWLIST`: Die Anwendung muss eine Anfrage an den Statusserver stellen, um zu bestätigen, dass das Siegel als verifiziert und somit gültig gelistet ist. Dieser Status wird verwendet, um zu zeigen, dass das Siegel überprüft wurde und keine Hinweise auf Ungültigkeit vorliegen. Ist das Siegel nicht gelistet, wird es in der Anwendung als ungültig markiert.
- Liste der enthaltenen Datenfelder (TLV-Objekte) mit folgenden Angaben:
    - Tag (Position)
    - optional?
    - Name des Datenfeldes
    - Beschreibungstext
    - maximale Länge (zu ignorieren beim Datentyp `date`)
    - Datentyp

Es werden nur Datenfelder nach den in Abschnitt 3.2 genannten obligatorischen Feldern beschrieben. Es sind daher nur Tags von 4 bis 254 (`0x04` bis `0xfe`) zu verwenden.

Folgende Standard ASN.1 Datentypen (X.680) stehen zur Verfügung:

- BOOLEAN
- INTEGER
- OCTET STRING
- UTF8String
- DATE
- DATE-TIME

Das Profil ist in Form eines XML-Datensatzes gemäß nachfolgendem Schema zu erstellen.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

    <!-- Definiert einen Typ für UUIDv4 ohne Bindestriche, ausschließlich in
Großbuchstaben -->
    <xs:simpleType name="UUIDv4Type">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9A-F]{32}"/>
        </xs:restriction>
    </xs:simpleType>

    <!-- Definiert einen Typ für Tags, beschränkt auf Werte von 4 bis 254 -->
    <xs:simpleType name="tagType">
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="4"/>
            <xs:maxInclusive value="254"/>
        </xs:restriction>
    </xs:simpleType>

    <!-- Definiert eine leikaID, welche eine oder mehrere 14-stellige numerische
Identifikationsnummern beschreibt, getrennt durch Semikolons. -->
    <xs:simpleType name="LeikaIDType">
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{14}(;\d{14})*"/>
        </xs:restriction>
    </xs:simpleType>

    <!-- Definiert ASN.1-basierte Standarddatentypen für die Attribute jedes
'entry'-Elements -->
    <xs:simpleType name="typeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="BOOLEAN" />
            <xs:enumeration value="INTEGER" />
            <xs:enumeration value="OCTET_STRING" />
            <xs:enumeration value="UTF8String" />
            <xs:enumeration value="DATE" />
            <xs:enumeration value="DATE-TIME" />
        </xs:restriction>
    </xs:simpleType>

    <!-- Definiert, ob und welche Art von Überprüfung gegen den Statusserver
erforderlich ist -->
    <xs:simpleType name="statusType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="NONE" />
            <xs:enumeration value="BLOCKLISTING" />
            <xs:enumeration value="ALLOWLISTING" />
        </xs:restriction>
    </xs:simpleType>

    <!-- Definiert den komplexen Typ für Einträge, der verschiedene Felder
umfasst -->
    <xs:complexType name="entryType">
        <xs:sequence>
            <xs:element name="name" type="xs:string" />
            <xs:element name="description" type="xs:string" />
            <xs:element name="length" type="xs:positiveInteger" minOccurs="0" />
            <xs:element name="type" type="typeType" />
            <xs:element name="defaultValue" type="xs:string" minOccurs="0" />
        </xs:sequence>
        <xs:attribute name="tag" type="tagType" use="required" />
        <xs:attribute name="optional" type="xs:boolean" />
    </xs:complexType>
```

```
    <!-- Definiert das Hauptelement 'profile', das die Datenstruktur für Profile
festlegt,
    die zur Erstellung optischer Siegel verwendet werden -->
    <xs:element name="profile">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="profileNumber" type="UUIDv4Type" />
                <xs:element name="profileName" type="xs:string" />
                <xs:element name="creator" type="xs:string" />
                <xs:element name="category" type="xs:string" minOccurs="0"/>
                <xs:element name="leikaID" type="LeikaIDType" minOccurs="0"
maxOccurs="1" />
                <xs:element name="statusIndicator" type="statusType"
minOccurs="0"/>
                <xs:element name="entry" type="entryType" minOccurs="1"
maxOccurs="251" />
            </xs:sequence>
        </xs:complexType>
        <xs:unique name="tagNo">
            <xs:selector xpath="entry" />
            <xs:field xpath="@tag" />
        </xs:unique>
    </xs:element>
</xs:schema>
```

4.1 Statusserver

Der Statusserver ermöglicht es, den Status bereits ausgestellter optisch verifizierbarer digitaler Siegel nachträglich zu modifizieren. Zu diesem Zweck ist die Bereitstellung eines öffentlich zugänglichen Statusservers notwendig. Dieser bietet eine Schnittstelle an, um Anfragen zur Änderung des Status eines digitalen Siegels zu empfangen. Ebenso verfügt er über eine Schnittstelle zur Abfrage des aktuellen Status digitaler Siegel. Der Status eines Siegels kann dabei entweder anzeigen, dass ein ausgestelltes digitales Siegel zurückgezogen und somit ungültig ist, oder dass es verifiziert und dadurch ausdrücklich gültig ist. Des Weiteren bietet die Struktur des Statusservers die Möglichkeit, zu einem späteren Zeitpunkt Erweiterungen zu implementieren, die die Unterstützung zusätzlicher Szenarien der Statusanpassung ermöglichen.

4.1.1 Profildefinition

Statusserverabfrage

Um den Statusserver nutzen zu können, ist eine Profildefinition erforderlich, die anzeigt, um welchen Typ der Anforderung es sich handelt. In der folgenden Liste sind die verfügbaren Arten von Statusserver-Abfragen aufgeführt. Diese Liste ist nicht abschließend und kann in neueren Versionen der Technischen Richtlinie um weitere Anwendungsszenarien erweitert werden.

`NONE` – Gemäß diesem Eintrag ist keine Überprüfung gegen einen Statusserver erforderlich. Es handelt sich dabei um den Standardzustand (Default).

`BLOCKLIST` – Dieser Eintrag bedeutet, dass eine Negativliste von ausgestellten Siegeln existiert, bei der das Prinzip gilt, dass ein Siegel solange als gültig betrachtet wird, bis es ausdrücklich zurückgezogen und in der Negativliste entsprechend hinterlegt ist. Um die Gültigkeit zu überprüfen, muss der Hashwert des digitalen Siegels gegenüber einem Statusserver abgeglichen werden. Wenn der Statusserver zurückmeldet, dass der Hashwert auf der Negativliste geführt wird, gilt das digitale Siegel als zurückgezogen und somit als ungültig. Dies muss klar und unmissverständlich in der prüfenden Anwendung dargestellt werden, um zu verdeutlichen, dass das ausgestellte digitale Siegel von der ausstellenden Behörde zurückgezogen wurde und somit ungültig ist. Ist kein Eintrag zu dem Hashwert vorhanden, hat lediglich die übliche Signaturprüfung zu erfolgen.

`ALLOWLIST` – Dieser Eintrag bedeutet, dass eine Positivliste von ausgestellten Siegeln existiert, bei der das Prinzip gilt, dass ein Siegel grundsätzlich als ungültig betrachtet wird, solange der Hashwert des entsprechenden Siegels nicht explizit auf einem Statusserver hinterlegt ist. Um die Gültigkeit zu überprüfen, muss der Hashwert des digitalen Siegels gegenüber einem Statusserver abgeglichen werden. Wenn der Statusserver zurückmeldet, dass der Hashwert nicht auf der Positivliste geführt wird, gilt das digitale Siegel als nicht explizit als gültig deklariert und somit als ungültig. Dies muss klar und unmissverständlich in der prüfenden Anwendung dargestellt werden, um zu verdeutlichen, dass das ausgestellte digitale Siegel von der ausstellenden Behörde nicht explizit als gültig deklariert wurde und somit ungültig ist.

4.1.2 Verarbeitung von Anfragen durch den Statusserver

4.1.2.1 Entgegennahme, Prüfung und Speicherung von Hashwerten digitaler Siegel durch den Statusserver

In diesem Abschnitt werden die Spezifikationen für die Entgegennahme, Prüfung und gegebenenfalls die Speicherung von Hashwerten digitaler Siegel durch den Statusserver erläutert. Der Server bietet dafür eine spezifische Schnittstelle an, die es ermöglicht, Hashwerte digitaler Siegel und zugehörige Informationen zu empfangen. Die eingehenden Daten werden anschließend überprüft und, sofern sie den Anforderungen entsprechen, in einer Datenbank gespeichert. Der Statusserver erwartet einen Request im JWT-Format mit spezifischen Informationen, wie sie im Abschnitt „4.1.3.1 Statusaktualisierung digitaler Siegel“ beschrieben sind.

Sobald der Statusserver einen entsprechend formatierten JWT Request erhalten hat, beginnt die sequenzielle Verarbeitung der Anfrage nach den folgenden Schritten:

1. Laden des Zertifikats: Anhand des im Request enthaltenen Signer Identifier und der Certificate Reference wird das Zertifikat, das sowohl für die Integritätssignatur des Requests als auch für die Erstellung der Siegelsignatur verwendet wurde, von der Zertifikatsverwaltung geladen. Nachdem dies erfolgt ist, wird eine Überprüfung seiner Gültigkeit vorgenommen.
2. Überprüfung der JWT-Signatur: Die Signatur des JWT wird mit dem öffentlichen Schlüssel, der aus dem zuvor geladenen Zertifikat extrahiert wurde, überprüft. Diese Überprüfung bestätigt, dass die übermittelten Daten authentisch und während der Übertragung unverändert geblieben sind.
3. Überprüfung der Siegelsignatur: Nach der erfolgreichen Verifizierung des JWT wird die Signatur des digitalen Siegels geprüft. Der im JWT enthaltene, Base64-kodierte Hashwert des Siegels („`hashValue`“) sowie der ebenfalls Base64-kodierte Signaturwert („`dssSigValue`“) werden zuerst dekodiert und dann für die Überprüfung herangezogen. Diese Überprüfung erfolgt unter Verwendung desselben Zertifikats, das bereits zur Validierung der Integritätssignatur des JWT verwendet wurde. Dies bestätigt nicht nur die Gültigkeit des digitalen Siegels, sondern gewährleistet auch, dass das Siegel und der Request mit demselben privaten Schlüssel signiert wurden, was eine Konsistenz zwischen Siegel und Anfrage sicherstellt.
4. Verarbeitung gemäß Anfragezweck („`statusPurpose`“): Sobald alle vorherigen Prüfungen erfolgreich abgeschlossen sind, wird der Request entsprechend seinem definierten Zweck verarbeitet:
   a Bei „`ADD`“: Der Hashwert des Siegels wird zusammen mit weiteren erforderlichen Informationen gespeichert. Dazu gehören die Art des Eintrags, wie beispielsweise `BLOCKLIST` oder `ALLOWLIST`, sowie der Gültigkeitszeitpunkt. Zusätzlich können weitere relevante Informationen erfasst und abgelegt werden, um eine umfassende Datenhaltung zu gewährleisten.
   b Bei „`REMOVE`“: Der zugehörige Eintrag des Siegels wird aus der Datenbank entfernt.
5. Senden einer Antwort: Nach der Verarbeitung des Requests muss der Statusserver eine Antwort im JSON-Format senden, die über die Schlüssel *status* und *message* den Ausgang der Anfrage bestimmt.

Die möglichen Werte für das *status*-Feld sind:

- „SUCCESS“: Die Operation (Hinzufügen oder Entfernen eines Hashwerts) wurde erfolgreich durchgeführt.
- „FAILURE“: Die Operation konnte nicht erfolgreich durchgeführt werden. Das *message*-Feld enthält Details zum Fehler.
- „ERROR“: Ein Verarbeitungsfehler ist aufgetreten, der unabhängig von der spezifischen Operation ist. Das *message*-Feld enthält Details zum Fehler.

4.1.2.2 Validierung eines digitalen Siegels über den Statusserver

Im Folgenden werden die Anforderungen für die Annahme und Überprüfung von Hashwerten durch den Statusserver erläutert. Der Server stellt eine spezifische Schnittstelle bereit, um Hashwerte digitaler Siegel sowie zugehörige Informationen entgegenzunehmen. Der Statusserver erwartet einen Request mit spezifischen Informationen, wie sie im Abschnitt „Statusabfrage digitaler Siegel“ beschrieben sind.

Nach dem Empfang einer entsprechend formatierten Anfrage verarbeitet der Statusserver diese in folgenden Schritten:

1. Identifizierung und Extraktion des Hashwerts: Der Server extrahiert den in der Anfrage übermittelten Hashwert, welcher das zu überprüfende digitale Siegel repräsentiert.
2. Datenbankabfrage: Mit dem extrahierten Hashwert führt der Server eine Abfrage in seiner Datenbank durch. Ziel ist festzustellen, ob der Hashwert bereits hinterlegt ist und welcher Status ihm zugeordnet ist. Dabei werden nicht nur die Existenz und das zugehörige Statuskennzeichen des Hashwerts überprüft, sondern auch ob der Gültigkeitszeitpunkt des Eintrags nicht überschritten ist.
3. Erstellung der Antwort: Der Statusserver erstellt eine Antwort im JSON-Format, die den Zustand des geprüften digitalen Siegels widerspiegelt. Die Antwort besteht aus zwei Hauptkomponenten: *status* und *message*. Mögliche Werte für das status-Feld sind:
   a „REVOKED“: Das digitale Siegel befindet sich auf der Blocklist und ist somit zurückgerufen worden.
   => Das digitale Siegel ist ungültig.
   b „NOT_REVOKED“: Das digitale Siegel ist nicht auf der Blocklist und gilt daher als nicht zurückgerufen.
   => Das digitale Siegel ist gültig.
   c „VERIFIED“: Das digitale Siegel ist auf der Allowlist und gilt somit als explizit verifiziert.
   => Das digitale Siegel ist gültig.
   d „UNVERIFIED“: Das digitale Siegel ist nicht auf der Allowlist und wurde daher nicht explizit verifiziert.
   => Das digitale Siegel ist ungültig.
   e „INVALID_CERT“ – Das verwendete Zertifikat ist unbekannt, zurückgerufen oder abgelaufen.
   => Das digitale Siegel ist ungültig.
   f „ERROR“: Ein Verarbeitungsfehler ist aufgetreten. Das *message*-Feld enthält Details zum Fehler.
   => Das digitale Siegel ist ungültig.
4. Übermittlung der Antwort: Die generierte Antwort wird an das anfragende Endgerät zurückgesendet, welches daraus den aktuellen Status des Siegels ableiten kann.

4.1.3 Anfrageerstellung für den Statusserver

4.1.3.1 Statusaktualisierung digitaler Siegel

Im Folgenden werden die Spezifikationen für die Übermittlung von Anfragen an den Statusserver erläutert. Um eine einheitliche Verarbeitung und Interoperabilität zu gewährleisten, müssen alle in diesem Abschnitt

festgelegten Vorgaben eingehalten werden. Für die Übertragung der Daten muss das JSON Web Token (JWT) (RFC7519) Format im Body eines HTTPS-POST-Requests verwendet werden. Der Statusserver muss auf Anfragen mit angemessenen http-Statuscodes reagieren, um den Erfolg oder spezifische Fehlerzustände zu signalisieren. Die Authentifizierung am Statusserver muss durch eine geeignete Authentifizierungsmethode erfolgen, um den Zugriff zu autorisieren, während das JWT die Integrität und die korrekte Herkunft der übermittelten Daten sicherstellt. Im Folgenden wird die Struktur des JWT beschrieben, die zur Übertragung verwendet wird.

JWT Header

Der JWT Header muss folgende Spezifikationen einhalten:

- `alg (Algorithm)`: Muss auf ES256 gesetzt werden. Dieser Wert ist verpflichtend und definiert den Einsatz von ECDSA mit P-256 und SHA-256 als Signaturalgorithmus.
- `typ (Type)`: Muss auf JWT gesetzt werden, um explizit das Format des Tokens zu spezifizieren.

JWT Payload

Der Payload des JWT muss alle nötigen Informationen zur Statusaktualisierung beinhalten. Die Felder sind wie folgt zu befüllen:

- *statusPurpose*: Gibt den Zweck der Anfrage an. Mögliche Werte:
  - `ADD`: Hinzufügen eines neuen Hashwerts.
  - `REMOVE`: Entfernen eines vorhandenen Hashwerts.
- *validityType*: Definiert die Art der Gültigkeit des Eintrags. Mögliche Werte:
  - `BLOCKLIST`: Der Eintrag markiert das Siegel als ungültig.
  - `ALLOWLIST`: Der Eintrag bestätigt die Gültigkeit des Siegels.
- *signerIdentifier*: Die eindeutige Identifikation des Signierers oder der Entität, die das Siegel signiert hat. Dies sollte gemäß den Spezifikationen in (DOC9303-13) erfolgen. Für optisch verifizierbare digitale Siegel zum Schutz von Verwaltungsdokumenten lautet dieser stets „`DEZV`“.
- *certificateReference*: Die Referenznummer des Zertifikats, das zur Signatur des Siegels verwendet wurde. Die Formatierung der UUID erfolgt als Großbuchstaben ohne Bindestriche gemäß (RFC4122).
- *hashValue*: Der Base64-kodierte Hashwert, welcher mittels des SHA-256-Algorithmus aus der Verkettung von Header und vollständiger Nachrichtenzone gebildet wird, unter Ausschluss des Tags, welcher den Beginn der Signaturzone oder die Länge der Signatur angibt, gemäß (DOC9303-13).
- *dssSigValue*: Der Base64-kodierte Signaturwert des digitalen Siegels, repräsentiert durch das im Rohformat vorliegende Wertepaar (r, s), formatiert als ASN.1-kodierte Sequenz gemäß (RFC3279).
- *validUntil*: Der spätestmögliche Zeitpunkt, bis zu dem der Eintrag als gültig betrachtet wird, formatiert im (ISO8601)Format. Dieser gibt an, wie lange der Eintrag in der Datenbank des Statusservers gültig und relevant bleibt. Dieser Mechanismus dient dazu, die Datenbank zu pflegen und Einträge nach ihrer Relevanz zu löschen. Die Dauer der Gültigkeit eines Eintrags darf nicht über die Gültigkeitsdauer des zugehörigen Zertifikats hinausgehen. Sollte kein spezifisches Ablaufdatum festgelegt sein, wird das Ablaufdatum des zugrundeliegenden Zertifikats als Standardwert angenommen.

JWT Signatur

Die Signatur des JWT muss unter Verwendung des ECDSA-Signaturalgorithmus mit SHA-256, spezifiziert als ES256, erzeugt werden. Diese Signatur muss die Base64URL-kodierte Darstellung des kombinierten Headers und Payloads absichern und am Ende des JWT angefügt werden. Zusätzlich muss die Signatur mit demselben privaten Schlüssel erstellt werden, der auch zur Generierung des Signaturwerts („`dssSigValue`“) des Siegels verwendet wurde. Dies gewährleistet, dass nur der berechtigte Aussteller des

digitalen Siegels, der im Besitz des entsprechenden privaten Schlüssels ist, Anfragen an den Statusserver übermitteln kann.

Serverantwort

Nach der Übermittlung des JWT-Requests antwortet der Statusserver mit einer strukturierten Antwort im JSON-Format, deren Spezifikationen im Abschnitt „4.1.2.1 Entgegennahme, Prüfung und Speicherung von Hashwerten digitaler Siegel durch den Statusserver“ beschrieben sind. Die Antwort umfasst den aktuellen Status des digitalen Siegels sowie etwaige Fehlerdetails.

4.1.3.2 Statusabfrage digitaler Siegel

Im Folgenden werden die Spezifikationen für die Übermittlung von Anfragen an den Statusserver erläutert. Um eine einheitliche Verarbeitung und Interoperabilität zu gewährleisten, sind alle in diesem Abschnitt festgelegten Vorgaben einzuhalten. Für die Übertragung der Daten ist das JSON-Format zu verwenden, welches im Body eines HTTPS-POST-Requests gesendet wird. Es wird erwartet, dass der Statusserver auf Anfragen mit angemessenen HTTP-Statuscodes reagiert, um den Erfolg oder spezifische Fehlerzustände zu signalisieren. Die Authentifizierung am Statusserver erfolgt durch eine geeignete Authentifizierungsmethode, um den Zugriff zu autorisieren. Die genaue Struktur der JSON-Daten, einschließlich aller erforderlichen und optionalen Felder, wird im Folgenden erläutert:

- Statuskennzeichen („`validityType`“)

  Definiert die Art der Gültigkeit des Eintrags, wobei folgende Werte möglich sind:

  - Zurückgezogen („`BLOCKLIST`“): Der Eintrag markiert das Siegel als ungültig.
  - Verifiziert („`ALLOWLIST`“): Der Eintrag bestätigt die Gültigkeit des Siegels.

- Hashwert („`hashValue`“):

  Der Hashwert wird mittels des SHA-256-Algorithmus aus der Verkettung von Header und vollständiger Nachrichtenzone gebildet, unter Ausschluss des Tags, welcher den Beginn der Signaturzone oder die Länge der Signatur angibt, gemäß (DOC9303-13).

Basierend auf der 4.1.2.2 Rückmeldung des Statusservers ist es Aufgabe der anfragenden Anwendung, nach der initialen Prüfung des digitalen Siegels eine abschließende Entscheidung über dessen Gültigkeit zu treffen. Diese Entscheidung basiert auf dem Vorhandensein des Hashwertes in der Datenbank des Statusservers sowie auf dem Kontext des Statuskennzeichens:

- Bei einem `ALLOWLIST` Statuskennzeichen: Die App erwartet, dass der Hashwert des Siegels in der Datenbank des Statusservers vorhanden ist. Nur wenn der Hashwert gefunden wird, gilt das Siegel als verifiziert und somit als gültig. Fehlt der Hashwert in der Datenbank, ist das Siegel nicht als verifiziert erfasst und wird als ungültig behandelt.
- Bei einem `BLOCKLIST` Statuskennzeichen: In diesem Fall bedeutet das Vorhandensein des Hashwertes in der Datenbank, dass das Siegel zurückgezogen und somit ungültig ist. Ist der Hashwert nicht in der Datenbank zu finden, kann davon ausgegangen werden, dass kein Widerruf vorliegt und das Siegel als gültig betrachtet werden kann.

Die Abfrage beim Statusserver ermöglicht es der App somit, den aktuellen Status eines Siegels zu ermitteln und sicherzustellen, dass Entscheidungen bezüglich der Gültigkeit des Siegels auf den neuesten Informationen basieren.

Literaturverzeichnis

DOC9303-13. ICAO: Doc 9303 Machine Readable Travel Documents – Part 13: Visible Digital Seals, Eigth Edition, March 2021.

eIDAS-VO. *Verordnung (EU) Nr. 910/2014 des Europäischen Parlaments und des Rates vom 23. Juli 2014 über elektronische Identifizierung und Vertrauensdienste für elektronische Transaktionen im Binnenmarkt und zur Aufhebung der Richtlinie 1999/93/EG.*

ISO8601. *Date and time format, 2019.*

RFC3279. *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, April 2002.*

RFC4122. A Universally Unique IDentifier (UUID) URN Namespace, July 2005.

RFC7519. *JSON Web Token (JWT), May 2015.*

TR-03137-1. Technical Guideline TR-03137: Optically Verifiable Cryptographic Protection of non-electronic Documents (Digital Seal).

X.680, ITU-T. *OSI networking and system aspects – Abstract Syntax Notation One (ASN.1): Specification of basic notation, 02/2021.*