



Checkliste für die Signaturerstellung
Dr. Peter Teufl
Version von 12.12.2016

```
"verificationTextualDescription" : "In diesem Modul und den dazugehörigen Submodulen werden die krypto  
"verificationState" : "FAIL",  
"verificationTimestamp" : "2016-10-27T10:37:16.579+02:00",  
"verificationResultList" : [ {  
  "verificationId" : "CRYPTO_SIGNATURE",  
  "version" : 1,  
  "verificationName" : "Kryptographie: Überprüfung der kryptographischen Gültigkeit der Signatur",  
  "verificationTextualDescription" : "In diesem Modul wird die kryptographische Gültigkeit der Signatur  
  "verificationState" : "FAIL",  
  "verificationTimestamp" : "2016-10-27T10:37:16.579+02:00"  
} ]  
} ]  
} ]
```

ÜBERSICHT

- [Signaturprüfung schlägt bei Prüfung mit Prüftool oder über FinOnline fehl
- [Diese Folien skizzieren eine Checkliste für die Unterstützung der Fehlersuche
- [Diese Folien ersetzen nicht die detaillierte Spezifikation! Bitte unbedingt das Dokument “Festlegungen des BMF zu Detailfragen der Registrierkassensicherheitsverordnung (RKSV)” beachten!
- [Achtung: Die Werte in den Beispiel-Screenshots sind nur demonstrativ und stellen keinen korrekten Beleg dar.

ÜBERSICHT

Checkliste, Überblick

1. Überprüfung der Daten in der CMC Datei
2. Aufbereitung der zu signierenden Daten
3. Hashwert-Berechnung
4. Signaturerstellung
5. Aufbereitung des Signaturwerts

AUSGANGSLAGE

- [Es wird davon ausgegangen, dass ein Beleg mit dem Prüftool überprüft wird und dabei der auf der ersten Folie beschriebene Fehler auftritt.
- [Für das Prüftool wird die CryptographicMaterialContainer-Datei (CMC) benötigt, die
 - die Zertifikate (oder direkt die öffentlichen Schlüssel bei einem GGS) und
 - den passenden AES-Schlüssel für die vorliegenden Belege enthält.
- [Die CMC-Datei wurde erstellt und das Prüftool wird damit verwendet: Dabei tritt der oben beschriebene Fehler tritt auf.
- [Wie kann die Fehlersuche nun effizient durchgeführt werden?

1. ÜBERPRÜFUNG DER DATEN IN DER CMC DATEI

Entspricht die Seriennummer in der CMC-Datei jener die im Beleg abgebildet ist?

Achtung: in V1.0.0. des Prüftools ist auch Groß/Kleinschreibung relevant (nicht in FinOnline)

```
{
  "base64AESKey" : "WQRtiiya3...",
  "certificateOrPublicKeyMap" : {
    "22166844ea8dccf2" : {
      "id" : "22166844ea8dccf2",
      "signatureDeviceType" : "CERTIFICATE",
      "signatureCertificateOrPublicKey" : "MIIBgE..."
    }
  }
}
```

_R1-AT100_CASHBOX-DEMO-1_CASHBOX-DEMO-1-Receipt-ID-1_2016-03-
11T03:57:08_0,00_0,00_0,00_0,00_0,00_4r1iIdZGeAQ=_22166844ea8dccf2_
cg8hNU5ihto=_W3QnrnkeXRQol6MUM84Qa8iXfhyWo8iQTorYfNjQp6LXNBWvIcp0Eq
YSg9ujIgXA2/0/rKs7SvQ21Yt+co9Ylg==

1. ÜBERPRÜFUNG DER DATEN IN DER CMC DATEI

Wird das passende Zertifikat in der Datei abgelegt?

Das BASE64-kodierte Zertifikat, das in der CMC-Datei unter der jeweiligen Seriennummer abgelegt wurde, muss jenen öffentlichen Schlüssel enthalten, der zu dem privaten Schlüssel (z.B. Karte, HSM) passt mit dem der Beleg signiert wurde.

2. AUFBEREITUNG DER ZU SIGNIERENDEN DATEN

Wie werden die Daten für die Hashwert-Berechnung aufbereitet?

Header

```
{"alg": "ES256"}
```

BASE64-URL Kodierung

```
eyJhbGciOiJIUzI1NiJ9
```

Payload

```
_R1-AT100_CASHBOX-DEMO-1_CASHBOX-DEMO-1-Receipt-ID-1_2016-03-11T03:57:08_0,00_0,00_0,00_0,00_0,00_4r1iIdZGeAQ=_22166844ea8dccf2_cg8hNU5iht0=
```

BASE64-URL Kodierung

```
X1IxLUFUMTAwX0NBU0hCT1gtREVNTy0xX0NBU0hCT1gtREVNTy0xLVJlY2VpcHQtsUQtMV8yMDE2LTAzLTExVDZ0jU30jA4XzAsMDBfMCwwMF8wLDAwXzAsMDBfMCwwMF80cjFpSWRaR2VBUT1fMjIxNjY4NDRLYThkY2NmMl9jZzhoTlU1aWh0bz0
```

Zusammenfügen mit "."

```
eyJhbGciOiJIUzI1NiJ9.X1IxLUFUMTAwX0NBU0hCT1gtREVNTy0xX0NBU0hCT1gtREVNTy0xLVJlY2VpcHQtsUQtMV8yMDE2LTAzLTExVDZ0jU30jA4XzAsMDBfMCwwMF8wLDAwXzAsMDBfMCwwMF80cjFpSWRaR2VBUT1fMjIxNjY4NDRLYThkY2NmMl9jZzhoTlU1aWh0bz0
```

Ergebnis:
Daten für die Hash-Berechnung

3. HASHWERT-BERECHNUNG

Wie wird der Hash-Wert berechnet?

header.payload aus JWS-Repräsentation

```
eyJhbGciOiJIUzI1NiJ9.X1IxLUFUMTAwX0NBU0hCT1gtREVNTy0xX0NBU0hCT1gtREVNTy0xLVJlY2VpcHQtsUQtMV8yMDE2LTAzLTExVDZjU3OjA4XzAsMDBfMCwwMF8wLDAwXzAsMDBfMCwwMF80cjFpSWRaR2VBUT1fMjI  
xNjY4NDRLYThkY2NmMl9jZzhoTlU1aWh0bz0
```



SHA-256 anwenden

Ergebnis: SHA-256 Hash-Wert (256 Bits, Byte-Array der Länge 32)

SHA256-HASH-WERT

4. SIGNATURERSTELLUNG

Wie wird die Signatur erstellt?

- Die Signaturerstellung erfolgt laut dem JWS-Standard. Wichtig dabei ist, dass das Ergebnis der Signaturerstellung, der Signaturwert, NICHT IM DER-Format kodiert wird, sondern einfach die beiden Resultate R und S zusammengefügt werden: Byte Array der Länge 64: R | | S. Das Zusammenfügen führt bereits die Signatureinheit durch.
- Typischerweise retournieren Smart-Card bereits dieses Ergebnis, es kann somit direkt weiterverwendet werden. Wenn das Ergebnis im DER-Format retourniert wird, muss die Umwandlung durchgeführt werden (siehe Muster-Code) (Schnell-Check: Hat das Ergebnis die Länge 64? Dann sehr wahrscheinlich dass R | | S Kodierung verwendet wird.)
- Bitte unbedingt bei der jeweiligen Lösung (HSM, Smardcard, Remote, aber auch Programmiersprache) darauf achten in welchem Format das Ergebnis retourniert wird. Dies wird in der jeweiligen Dokumentation genannt, bzw. kann der VDA weiterhelfen.
- ACHTUNG: Die Signaturerstellung benötigt immer den Hash-Wert und niemals die Plain-Text Daten (also in diesem Fall die HEADER.PAYLOAD Repräsentation). Es kann aber sein, dass ein API dies vereinfacht und nur die Plain-Text Daten übergeben werden. Die Hash-Wert Berechnung erfolgt dann vor der Signaturerstellung intern und ist nicht ersichtlich. Dazu bitte die Dokumentation des jeweiligen APIs lesen oder den VDA fragen.

Signierung des Hash-Werts

SHA256-HASH-WERT



Sig-Wert: R | | S

5. AUFBEREITUNG DES SIGNATURWERTS

Wie wird der Signaturwert für die Darstellung im maschinenlesbaren Code aufbereitet?

BASE64-Kodierung

Ablegen als letztes Element im maschinenlesbaren Code

Sig-Wert: R | | S

BASE64-Kodierung

BASE64-SigWert

_R1-AT100_CASHBOX-DEMO-1_CASHBOX-DEMO-1-Receipt-ID-1_2016-03-11T03:57:08_0,00_0,00_0,00_0,00_0,00_4r1iIdZGeAQ=_22166844ea8dccf2_cg8hNU5ihto=

BASE64-SigWert

_R1-AT100_CASHBOX-DEMO-1_CASHBOX-DEMO-1-Receipt-ID-1_2016-03-11T03:57:08_0,00_0,00_0,00_0,00_0,00_4r1iIdZGeAQ=_22166844ea8dccf2_cg8hNU5ihto=_W3QnrnkeXRQol6MUM84Qa8iXfhyWo8iQToRyfnjQp6LXNBWvIcp0EqYSg9ujIgXA2/0/rKs7SvQ21Yt+co9Ylg==

5. AUFBEREITUNG DES SIGNATURWERTS

Sig-Wert: R | | S

Wie wird der Signaturwert für die Darstellung in der JWS-Repräsentation für das RKSVD-EP (und auch die Berechnung der Verkettung) aufbereitet?

BASE64URL-Kodierung



BASE64URL-SigWert

- BASE64URL-Kodierung
- Ablegen als letztes Element im maschinenlesbaren Code

header.payload

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLT1gtREVNTy0xX0NBUEhCT1gtREVNTy0xLVJlY2VpcHQtsUQtMV8yMDE2LTAzLTExVDZ0jU30jA4XzAsMDBfMCwwMF8wLDAwXzAsMDBfMCwwMF80cjFpSWRaR2VBUT1fMjI  
xNjY4NDRLYThkY2NmMl9jZzhoTlU1aWh0bz0
```

signature

BASE64URL-SigWert

Zusammenfügen mit "."

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLT1gtREVNTy0xX0NBUEhCT1gtREVNTy0xLVJlY2VpcHQtsUQtMV8yMDE2LTAzLTExVDZ0jU30jA4XzAsMDBfMCwwMF8wLDAwXzAsMDBfMCwwMF80cjFpSWRaR2VBUT1fMjI  
xNjY4NDRLYThkY2NmMl9jZzhoTlU1aWh0bz0
```

kompakte JWS-Rep:
header.payload.signature

5. AUFBEREITUNG DES SIGNATURWERTS

Unbedingt beachten:

- Die JWS-Repräsentation, die für die Berechnung des Hashwerts und im DEP-Export verwendet wird, schreibt die BASE64-URL Kodierung vor. Im JWS-Standard werden aber "=" Padding-Zeichen ausgeschlossen. (Diese sind im BASE64-URL Standard an sich erlaubt).
- Wenn Sie die Meldung einer ungültigen Signatur im Prüftool erhalten, bitte unbedingt beachten ob in der Aufbereitung für die Berechnung des Hashwerts (header.payload) BASE64-URL Kodierung ohne "=" Padding verwendet wird. Ist dies der Fall, dann müssen die "=" Zeichen entfernt werden.