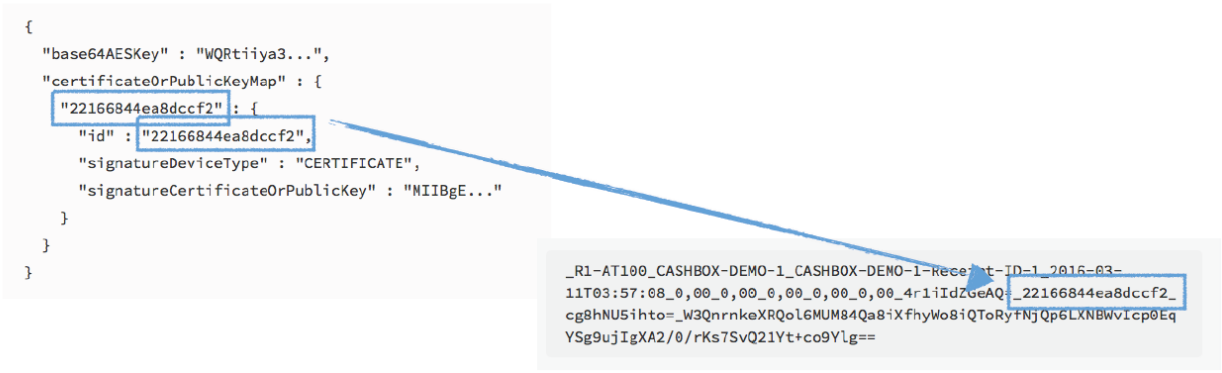
1. ÜBERPRÜFUNG DER DATEN IN DER CMC DATEI

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)



Check the serial numbers used in

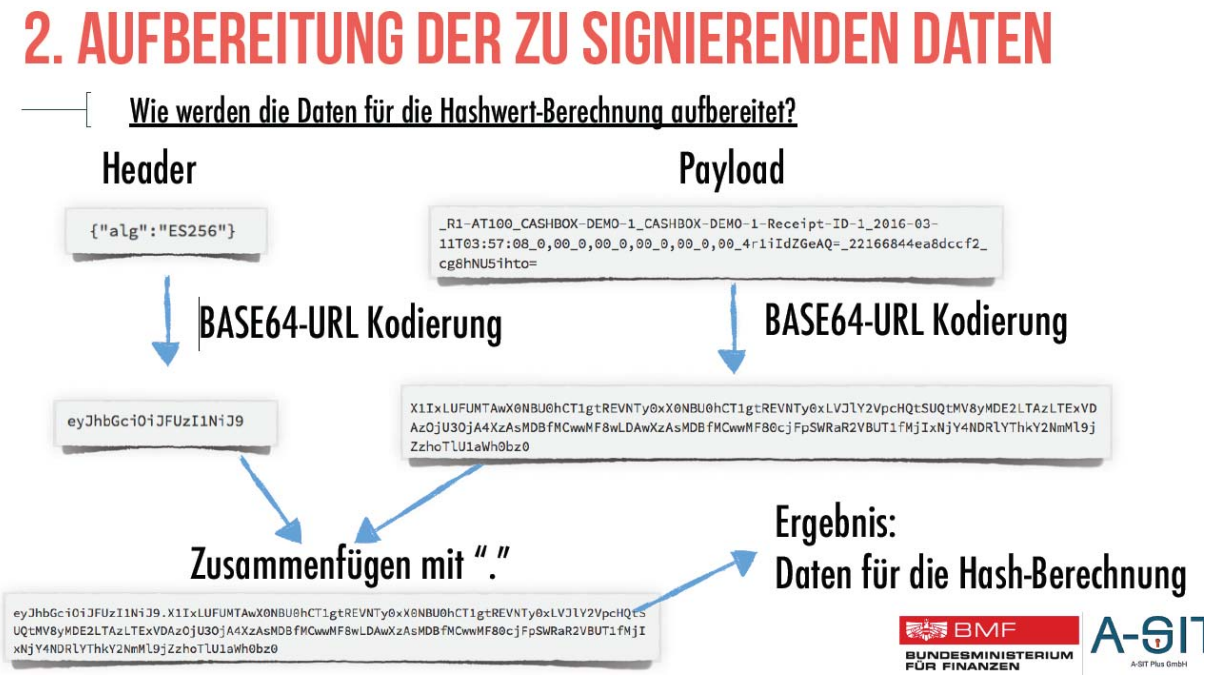
- cryptographicMaterialContainer
- machine readable code
- for the A-SIT-Tool the serial number is case sensitive

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.

Does the certificate info in cryptographicMaterialContainer contain the public key which matches the private key used for signing?

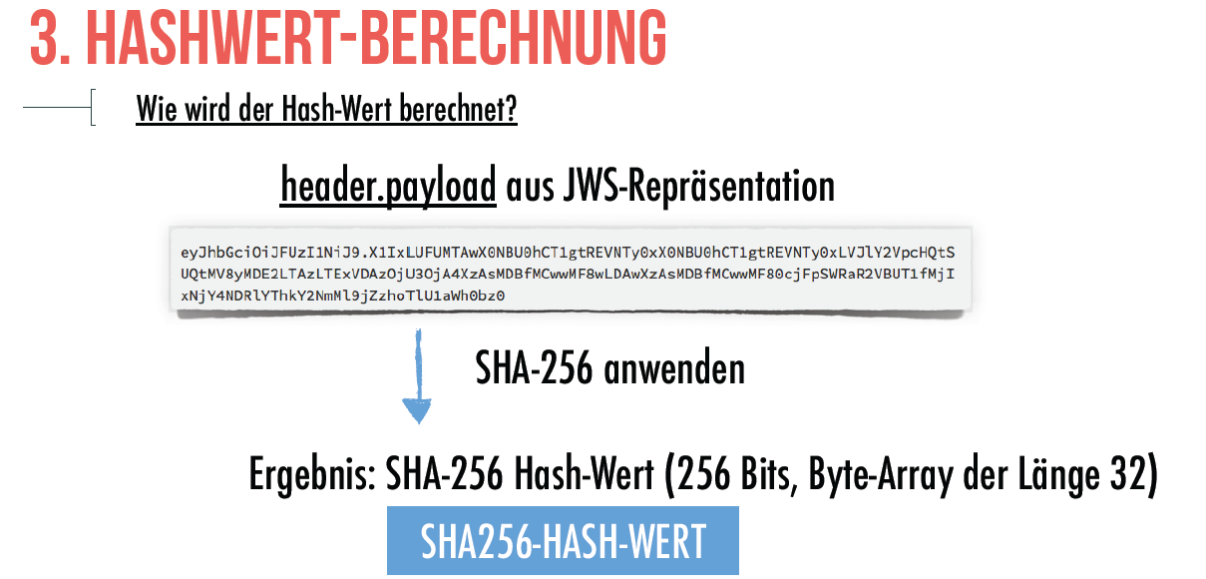


Take header and payload

Do BASE64URL coding

Concatenate them with „.“

Result ist input for hash



Calc the Hash with SHA-256, Output is 32 bytes

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.

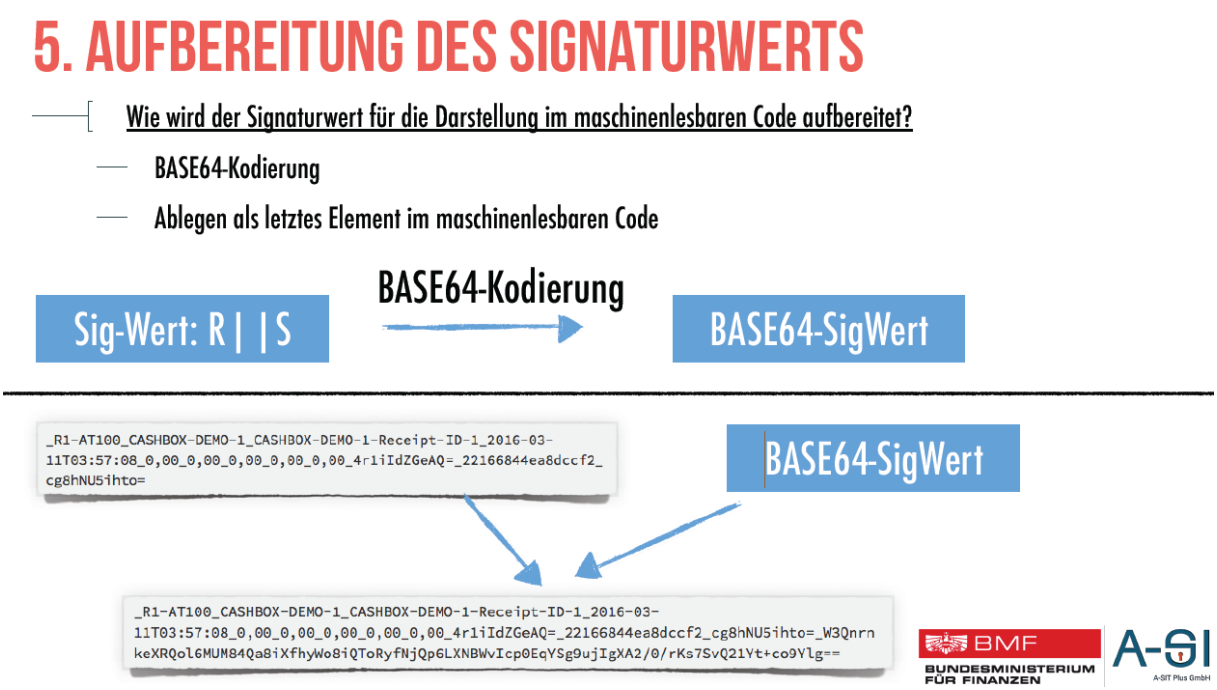
Check signature process:

Signature generation is done by JWS standard. The result is R and S. This is wrong if DER coded! Result is a byte array with 64 bytes. Concatenation of R and S is normally done by the signature unit.

If the result is DER coded it has to be converted. Fast check: if length = 64 then it normally will be R and S coded.

It depends on the format of the result – check smartcard interface documentation, program language etc.

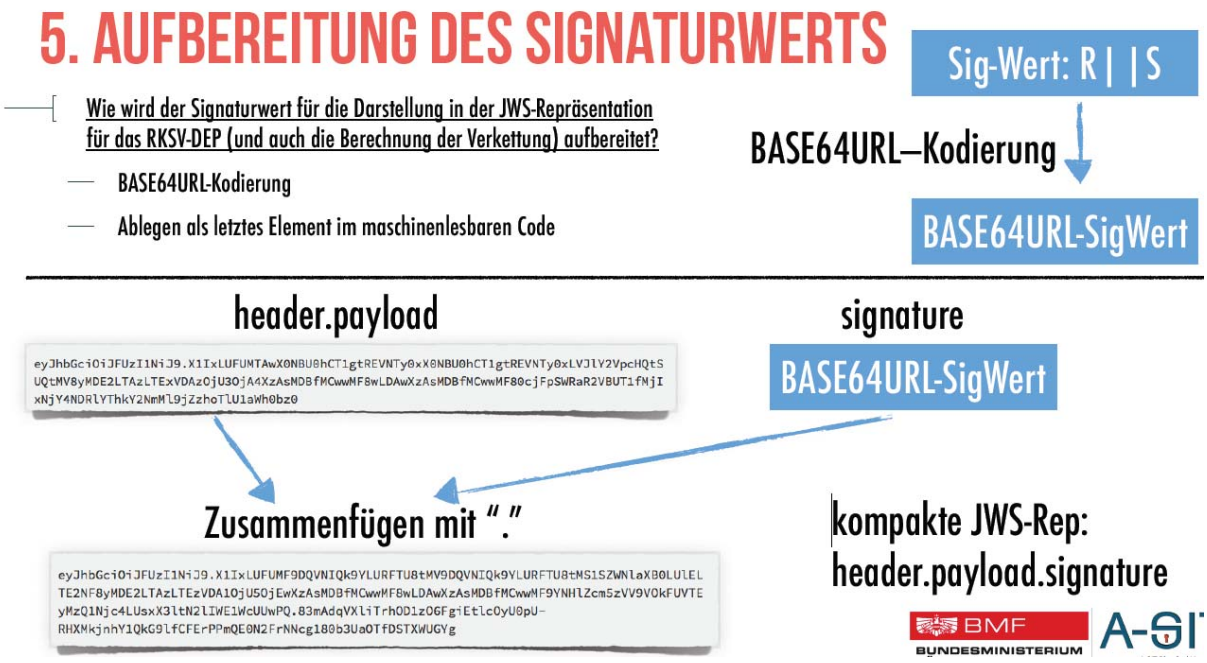
The signature process itself always needs the hash value, not the plain data. In some card interfaces the API simplifies this and requests instead the plain data (and does the hash internally) – check smartcard interface documentation!



Process for machine readable code:

Convert signature result (R and S) to BASE64 value.

Add as last element to the machine readable code.



For the DEP export use BASE64URL coding of the signature value!

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.

In the DEP export format the character „=“ is not allowed – remove any of them.