Authentisierung Aktensystem als Client der VST

Nun sind die Schlüssel K2_c2s_AppData und K2_s2c_AppData erfolgreich erzeugt.

Das AS erzeugt die Daten

```
GET /epa/authz/v1/freshness HTTP/1.1
Host: www.vst...
```

Das wird mittels K2_c2s_AppData und AES/GCM verschlüsselt (https://gemspec.gematik.de/docs/gemSpec/gemSpec_Krypt/latest/#A_24632).

Die Anwendungslogik in der VST entschlüsselt den Request und erzeugt eine Nonce. Antwort dann mit K2_s2c_AppData verschlüsselt. Entschlüsselte Anwort dann bspw.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Date: Fri, 21 Jun 2024 14:18:33 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Content-Length: xxx

xxxx korrigieren
YSBiYXNlNjQgZW5jb2RlZCBjaGFsbGVuZ2UgKGhlYWx0aCByZWNvcmQgc3lzdGVtIHNwZWNpZmljIGNvbnRlbnQp
```

Dann erzeugt das Aktensystem einen JWT-Body

```
{
   "type" : "ePA-Authentisierung über PKI",
   "iat" : ...zeit...,
   "challenge" :
"YSBiYXNlNjQgZW5jb2RlZCBjaGFsbGVuZ2UgKGhlYWx0aCByZWNvcmQgc3lzdGVtIHNwZWNpZmljIGNvbnRlbnQp",
   "sub": "9-ePA-AS"
```

```
}
```

Das signiert das AS mittels seine VAUAUT Schlüsselmaterials und erzeugt damit einen JWT ganz klassisch in compact form. Dann erzeugt das AS eine Zeichenkette

```
POST /epa/authz/v1/send_authorization_request_bearertoken HTTP/1.1
Host: www.vst...
Content-Type: application/json; charset=UTF-8
Content-Length: xxy

34893028409332sljdaslkjdlaksjdlasjaljdas.e07radasdlkjasldjsaldjalsdjas.sajdlasue908ejdlskajdlaksjd
```

Das wird mittel K2_c2s_Data verschlüsselt und an die Anwendungslogik gesendet. Die Anwendungslogik entschlüsselt das mittels K2_c2s_Data. Extrahiert das JWT und prüft das JWT (insbesondere die Signatur) wie in

https://gemspec.gematik.de/docs/gemSpec/gemSpec_Krypt/latest/#A_24658-01 definiert.

Revision #4
Created 24 July 2025 10:41:59 by Admin
Updated 20 April 2026 06:19:50 by Admin