C_11865, Änderungen in gemSpec_Krypt

ALT:

**A_15751-02 - TLS-Verbindung zwischen ePA-Aktensystem und ePA-Client**
Ein ePA-Aktensystem und ein ePA-Client MÜSSEN in Bezug auf die TLS-Verbindung zwischen ihnen

1. folgende Ciphersuiten unterstützen
   - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xC0, 0x30),
   - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xC0, 0x2F),
   - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xC0, 0x2C),
   - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xC0, 0x2B).
2. Sie KÖNNEN weitere Cipher-Suiten aus [TR-02102-2, Abschnitt 3.3.1 Tabelle 1] unterstützen.
3. Bei dem ephemeren Elliptic-Curve-Diffie-Hellman-Schlüsselaustausch und bei der Signaturprüfung mittels ECDSA MÜSSEN die Kurven P-256 oder P-384 [FIPS-186-5] unterstützt werden. Daneben SOLLEN die Kurven brainpoolP256r1, brainpoolP384r1 oder brainpoolP512r1 (vgl. [RFC-5639] und [RFC-7027]) unterstützt werden. Andere Kurven SOLLEN NICHT verwendet werden (Hinweis: die Intention des letzten Satzes ist insbesondere, dass die Ordnung des Basispunktes in E(F_p) nicht zu klein werden darf).

**[**<=, , **]**

NEU:

**A_15751-03 - TLS-Verbindung zwischen ePA-Aktensystem und ePA-Client**
Ein ePA-Aktensystem und ein ePA-Client MÜSSEN in Bezug auf die TLS-Verbindung zwischen ihnen

1. folgende Ciphersuiten unterstützen
   - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xC0, 0x2C),
   - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xC0, 0x2B).
2. Sie KÖNNEN weitere Cipher-Suiten aus [TR-02102-2, Abschnitt 3.3.1 Tabelle 1] unterstützen.
3. Bei dem ephemeren Elliptic-Curve-Diffie-Hellman-Schlüsselaustausch und bei der Signaturprüfung mittels ECDSA MÜSSEN die Kurven P-256 oder P-384 [FIPS-186-5] unterstützt werden. Daneben KÖNNEN die Kurven brainpoolP256r1, brainpoolP384r1 oder brainpoolP512r1 (vgl. [RFC-5639] und [RFC-7027]) unterstützt werden. Andere Kurven SOLLEN NICHT verwendet werden (Hinweis: die Intention des letzten Satzes ist insbesondere, dass die Ordnung des Basispunktes in E(F_p) nicht zu klein werden darf).

**[**<=, Aktensystem_ePA, CS_ePA_KTR, PS, PS_ePA_Apotheke, CS_ePA_DiGA, PS_ePA, CS_ePA_Ombudsstelle, NCPeH_FD, Frontend_Vers_ePA, funkt. Eignung: Herstellererklärung, funkt. Eignung: Konformitätsbestätigung, Sich.techn. Eignung: Produktgutachten**]**

Neue Anforderung nach A_15751-03

**A_26025 - ePA: TLS-Identitäten, IOP, P-256 Basierte ECC-Schlüssel**
Ein ePA-Aktensystem MUSS sicherstellen, dass seine TLS-Identitäten (vgl. A_15751-*) auf der Kurve P-256 [FIPS-186-5] basieren. D. h., der EE-Schlüssel im TLS-Server-Zertifikat als auch das bestätigende CA-Zertifikat (Komponenten-PKI-CA-Zertifikat) MÜSSEN als öffentliche Schlüssel Kurvenpunkte auf der ECC-Kurve P-256 besitzen.**[**<=, Aktensystem_ePA, funkt. Eignung: Herstellererklärung, Sich.techn. Eignung: Herstellererklärung**]**

Erläuterung:

Ziel ist es die Interoperabilität zwischen Standard-TLS-Bibliotheken/Security-Providern auf den Primärsystemen und der TLS-Implementierung/Konfiguration ePA-Aktensystem im Kontext des TLS-Verbindungsaufbaus mit dem ePA-Aktensystem sicherzustellen. Es wird mit A_26025-* gefordert, dass die Kurvenparameter P-256 für die TLS-Server-Schlüssel der ePA-Aktensysteme verwendet werden.

Ein Aktensystem-Betreiber erzeugt wie üblich die CSR für die TLS-Zertifikat des vom ihm betriebenen Aktensystems -- nun mit den ECC-Schlüsseln auf Basis von P-256 -- und übergibt diese per TMS an die Komponenten-PKI zur Zertifikatserstellung. Die Komponenten-PKI prüft den CSR und wählt automatisch eine Komponenten-CA, die auf P-256 basiert, als bestätigende Instanz (vgl. auch A_23139-*).

ALT:

**A_24425 - VAU-Protokoll: VAU-Schlüssel für die VAU-Protokoll-Schlüsselaushandlung**
Ein Aktensystem MUSS sicherstellen, dass

1. es eine Signatur-Identität aus der Komponenten-PKI der TI gibt, die technisch sichergestellt ausschließlich nur von VAU-Instanzen verwendbar ist (AUT-Zertifikat und Schlüsselmaterial (VAU-HSM) wie bei ePA 1x. und 2.x).
2. es semi-statische Schlüsselpaare für ECDH (auf Basis Kurve P-256 [FIPS-186-5]) und Kyber768 [IEFT-Kyber] gibt, deren private Schlüssel, technisch sichergestellt, ausschließlich von VAU-Instanzen verwendbar sind.
3. die privaten Schlüssel in einer VAU-Instanz erzeugt und verarbeitet werden (also nicht im VAU-HSM),
4. die semi-statischen Schlüssel eine maximale Lebensdauer von einem Monat besitzen (Hinweis: die Forward-Secrecy hängt nicht vom Wechselintervall ab, innerhalb eines Verbindungsaufbaus und der Schlüsselaushandlung dabei fließen ephemere Schlüsselwerte von Client und Sever ein).
5. die semi-statischen Schlüssel in einer über die Signatur-Identität authentisierten folgenden Datenstruktur aufgeführt werden.

**Struktur der signierten semi-statischen öffentlichen VAU-Schlüssel**

```
VAU_Keys = {
          "ECDH_PK" :
            { "crv" : "P-256",
              "x" : Binärwert-x-Koordinate-32-Bit-big-endian,
              "y" : Binärwert-x-Koordinate-32-Bit-big-endian,
            },
          "Kyber768_PK" : Binärwert-öffentlicher-Schlüssel-nach-keygen-
```

```
Spec-Kyber768,
          "iat" : Erzeugungszeits-Sekunden-Since-Epoch (integer),
          "exp" : Nicht-mehr-Verwendbar-nach (integer),
          "comment" : "Erzeugt bei VAU-Instanz xyz, Meta-Info abcd"
}
```

In "comment" KÖNNEN beliebige Text-Daten aufgeführt werden. Es können weitere Attribute hinzugeführt werden. Ein Client MUSS ihm unbekannte Attribute ignorieren. Diese Struktur wird mittels CBOR [RFC-CBOR] binär kodiert und im Folgenden VAU_Keys_encoded genannt.

Diese binäre Byte-Folge wird in folgende Datenstruktur eingebracht

```
{
 "signed_pub_keys" : VAU_Keys_encoded,
 "signature-ES256" : ECDSA-Signatur-SHA-256-analog-RFC-7515 (R||S => 64
Byte) binär,
 "cert_hash"       : SHA-256-Wert des "signierenden" AUT-VAU-Zertifikats,
 "cdv"             : Cert-Data-Version (natürliche Zahl, beginnend mit 1,
vgl. A_24957-*),
 "ocsp_response"   : OCSP-Response-für-das-VAU-Signaturzertifikat-nicht-
älter-als-24-Stunden-DER-Kodierung
}
```

Diese Datenstruktur wird mittels CBOR binär kodiert (serialisiert). Das Ergebnis der Kodierung wird "signierte öffentliche VAU-Schlüssel" (Plural) genannt.

**[<=, , ]**

NEU: (32 Bit -> 32 Byte)

**A_24425-01 - VAU-Protokoll: VAU-Schlüssel für die VAU-Protokoll-Schlüsselaushandlung**

Ein Aktensystem MUSS sicherstellen, dass

1. es eine Signatur-Identität aus der Komponenten-PKI der TI gibt, die technisch sichergestellt ausschließlich nur von VAU-Instanzen verwendbar ist (AUT-Zertifikat und Schlüsselmaterial (VAU-HSM) wie bei ePA 1x. und 2.x).
2. es semi-statische Schlüsselpaare für ECDH (auf Basis Kurve P-256 [FIPS-186-5]) und Kyber768 [IEFT-Kyber] gibt, deren private Schlüssel, technisch sichergestellt, ausschließlich von VAU-Instanzen verwendbar sind.
3. die privaten Schlüssel in einer VAU-Instanz erzeugt und verarbeitet werden (also nicht im VAU-HSM),
4. die semi-statischen Schlüssel eine maximale Lebensdauer von einem Monat besitzen (Hinweis: die Forward-Secrecy hängt nicht vom Wechselintervall ab, innerhalb eines Verbindungsaufbaus und der Schlüsselaushandlung dabei fließen ephemere Schlüsselwerte von Client und Sever ein).
5. die semi-statischen Schlüssel in einer über die Signatur-Identität authentisierten folgenden Datenstruktur aufgeführt werden.

**Struktur der signierten semi-statischen öffentlichen VAU-Schlüssel**

```
VAU_Keys = {
          "ECDH_PK" :
            { "crv" : "P-256",
              "x" : Binärwert-x-Koordinate-32-Byte-big-endian (256 Bit),
              "y" : Binärwert-x-Koordinate-32-Byte-big-endian (256 Bit),
            },
          "Kyber768_PK" : Binärwert-öffentlicher-Schlüssel-nach-keygen-
Spec-Kyber768,
          "iat" : Erzeugungszeits-Sekunden-Since-Epoch (integer),
          "exp" : Nicht-mehr-Verwendbar-nach (integer),
          "comment" : "Erzeugt bei VAU-Instanz xyz, Meta-Info abcd"
}
```

In "comment" KÖNNEN beliebige Text-Daten aufgeführt werden. Es können weitere Attribute hinzugeführt werden. Ein Client MUSS ihm unbekannte Attribute ignorieren. Diese Struktur wird mittels CBOR [RFC-CBOR] binär kodiert und im Folgenden VAU_Keys_encoded genannt.

Diese binäre Byte-Folge wird in folgende Datenstruktur eingebracht

```
{
 "signed_pub_keys" : VAU_Keys_encoded,
 "signature-ES256" : ECDSA-Signatur-SHA-256-analog-RFC-7515 (R||S => 64
Byte) binär,
 "cert_hash"       : SHA-256-Wert des "signierenden" AUT-VAU-Zertifikats,
 "cdv"             : Cert-Data-Version (natürliche Zahl, beginnend mit 1,
vgl. A_24957-*),
 "ocsp_response"   : OCSP-Response-für-das-VAU-Signaturzertifikat-nicht-
älter-als-24-Stunden-DER-Kodierung
}
```

Diese Datenstruktur wird mittels CBOR binär kodiert (serialisiert). Das Ergebnis der Kodierung wird "signierte öffentliche VAU-Schlüssel" (Plural) genannt.

**[**<=, Aktensystem_ePA, CS_ePA_KTR, PS_ePA_Apotheke, CS_ePA_DiGA, PS_ePA, CS_ePA_Ombudsstelle, NCPeH_FD, Frontend_Vers_ePA, funkt. Eignung: Herstellererklärung, Sich.techn. Eignung: Produktgutachten, funkt. Eignung: Konformitätsbestätigung**]**

## Änderungen in gemProdT_..._PTVx.y.z-n

*Anmerkung: Die Anforderungen der folgenden Tabelle stellen einen Auszug dar und verteilen sich innerhalb der Tabelle des Originaldokuments [gemProdT_...]. Alle Anforderungen der Tabelle des Originaldokuments, die in der folgenden Tabelle nicht ausgewiesen sind, bleiben unverändert bestehenden.*

**Tabelle 1: Anforderungen zur funktionalen Eignung "Produkttest/Produktübergreifender Test"**

| Afo-ID | Afo-Bezeichnung | Zuweisung |
|---|---|---|
| A_26025 | ePA: TLS-Identitäten, IOP, P-256 Basierte ECC-Schlüssel | Herstellererklärung: funktionale Eignung und |

| | | |
|---|---|---|
| | | sicherheitstechnische Eignung |