IRD Technische Spezifikation für meldende Gesundheitseinrichtungen (API-Version 2.0) - RFC

Dokumenteninformation

Version	Datum	Grund der Änderung, besondere Hinweise	Bearbeiter
1.0	16.04.2024	Initialversion	RKI Referat VIG (Vertrauensstelle-ird@rki.de)

Inhaltsverzeichnis

Einleitung

Diese technische Spezifikation beschreibt die Kommunikation einer meldenden Gesundheitseinrichtung über die REST-Schnittstelle der Vertrauensstelle IRD für die Meldungen zu einer implantatbezogenen Maßnahme nach IRegG einschließlich Korrektur und Stornierung dieser Meldungen, sowie der Initiierung eines Auskunftsverlangens entsprechend DSGVO zu den Daten eines Patienten, der im Implantateregister Deutschland erfasst ist.

Die technische Spezifikation richtet sich in erster Linie an Softwarehersteller, die die Implementierung der Anbindung an die Schnittstelle in Systemen wie z.B. Krankenhausinformationssysteme (KIS) oder Praxisverwaltungssysteme (PVS) implementieren.

Inhaltliche Pflichtangaben sind Felder, die sich mindestens aus §17 Abs. 1 IRegG (siehe https://www.gesetze-im-internet.de/iregg/__17.html) begründen sowie weitere notwendige technische Felder.

Allgemeine Informationen

Umgebungen

Die Vertrauensstelle ist innerhalb der Referenzumgebung (RU) der Telematikinfrastruktur unter https://vst-ird-ru.rki-ti.de zu erreichen und innerhalb der Produktivumgebung (PU) der Telematikinfrastruktur unter https://vst-ird.rki-ti.de.

Authentisierung, Authentifizierung und Autorisierung

Für die Benutzung der beschriebenen APIs müssen sich meldende Gesundheitseinrichtungen an der API mithilfe eines von der Registerstelle ausgestellten, signierten JWT-Token anmelden. Die Herausgabe dieses signierten JWT-Tokens geschieht durch die Anmeldung der Gesundheitseinrichtung an den IDP der gematik sowie der Weiterleitung des ausgestellten Tokens an den IDP der Registerstelle. Weitere Informationen dazu finden sich in den Dokumentation der Registerstelle unter HB_LeitfadenRegistrierung sowie unter Technische_Dokumentation v2.1.1.

Die API Version 2.0 akzeptiert als patienten-identifizierendes Datum den unveränderlichen Teil der Krankenversichertennummer (KVNR) nach § 290 Absatz 1 Satz 2 SGB V oder eine andere eindeutige, unveränderbare und nach einheitlichen Kriterien gebildete Identifikationsnummer nach § 17 Absatz 4 Satz 3 IRegG[1]. Aktuell ist für diese Identifikationsnummer als Format 11-stellige Ziffernfolge festgelegt.

Datenformat

Eingabeformat

Die Daten müssen in dem Format **application/json** zu der Schnittstelle gesendet werden. Die Zeichenkodierung wird als **UTF-8** gelesen. Namen der Eigenschaften werden CaseInsensitive ausgewertet.

> **Festlegungen zu den JSON-Daten**
>
> - Es werden keine Kommentare in den JSON-Daten akzeptiert. Die Verwendung von Kommentaren führt zu einem Fehler.
> - Es werden keine doppelten Schlüssel (also Namen der Eigenschaften) in einem JSON-Knoten akzeptiert. Die Verwendung von doppelten Schlüsseln führt zu einem Fehler.
> - Es werden nur Eigenschaften akzeptiert, die auch in der OpenAPI-Definition definiert sind. Die Verwendung anderer, uns unbekannter Eigenschaften, führt zu einem Fehler.

Rückgabe

Nach erfolgreicher Verarbeitung der Daten wird der StatusCode 200 sowie die Antwort als Content zurückgegeben. Content-Type der Antwort ist **application/json** mit der Zeichencodierung **UTF-8**.

Signierung von Daten

Das Verfahren zur Signaturberechnung sowie die konkreten Formate, die benutzt werden müssen, sind aktuell in Klärung.

Signierung der Eingangsdaten

Signaturen der Daten, die von der Gesundheitseinrichtung mit den jeweiligen Daten mitzusenden sind, gewährleisten die Integrität und die Authentizität der übertragenenen Daten. Es wird hier das Schema Encrypt-Then-Sign angewendet, die Signatur wird also über die bereits verschlüsselten Daten gebildet.

Bei der Signatur handelt es sich um eine nonQES - Signatur (nicht qualifizierte elektronische Signatur).

Bei der Signatur werden die Eingangsdaten in einer konkatenierten Form zur Signaturberechnung und -prüfung betrachtet.

Für beispielsweise einem JSON-Fragment mit diesen Daten

```
"Patientendaten": {
   "IdVersicherter":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZFZlcnNpY2hlcnRlcg==",
   "IdDatensatz":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZERhdGVuc2F0eg==",
   "IdKrankenversicherung":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlciBJZEtyYW5rZW52ZXJzaWNoZXJ1bmc="
}
```

wird folgende Zeichenkette erstellt:

```
<Wert des IdVersicherten> + | + <Wert des IdDatensatzes> + | + <Wert der IdKrankenversicherung>

Hexadezimale Darstellung
49 63 68 20 62 69 6E 20 65 69 6E 20 76 65 72 73 63 68 6C C3 BC 73 73 65 6C 74 65 72 20 57 65 72 74 20 64 65 73
20 49 64 56 65 72 73 69 63 68 65 72 74 65 72 7C 49 63 68 20 62 69 6E 20 65 69 6E 20 76 65 72 73 63 68 6C C3 BC
73 73 65 6C 74 65 72 20 57 65 72 74 20 64 65 73 20 49 64 44 61 74 65 6E 73 61 74 7A 7C 49 63 68 20 62 69 6E 20
65 69 6E 20 76 65 72 73 63 68 6C C3 BC 73 73 65 6C 74 65 72 20 57 65 72 74 20 64 65 72 20 49 64 4B 72 61 6E 6B
65 6E 76 65 72 73 69 63 68 65 72 75 6E 67
```

Die Plain Bytes der Zeichenkette müssen dann an die entsprechende Schnittstelle des Konnektors bzw. des Basis-/KTR-Consumer gesendet werden. Nach der Operation der Signaturbildung wird dann die Signatur zurückgegeben. Diese ist dann in der Eigenschaft "Signatur" zu verwenden. Erweiterte Information zur Verwendung der Schnittstelle des Konnektors bzw. des Basis-/KTR-Consumer zur Signaturbildung und zu den konkreten Formaten finden sich unter dem Punkt Übergreifene Beispiele.

Das jeweilige Beispiel zur Bildung der Zeichenkette, die zur Signaturbildung verwendet werden muss, findet sich unter den jeweiligen API.

Signierung der Ausgangsdaten

Signaturen der Daten, die von der Vertrauensstelle mit der jeweiligen Antwort mitgesendet werden, gewährleisten die Integrität und die Authentizität der übertragenenen Daten.

Bei der Signatur handelt es sich um eine nonQES - Signatur (nicht qualifizierte elektronische Signatur). Die Signatur wird über die Daten der Antwort gebildet.

Erweiterte Information zur Verwendung der Schnittstelle des Konnektors bzw. des Basis-/KTR-Consumer zur Prüfung der Signatur und zu den konkreten Formaten finden sich unter dem Punkt Übergreifene Beispiele.

> Die von der Vertrauensstelle ausgestellte Signatur ist vor der Verarbeitung oder Weiterleitung von Daten an die Registerstelle zu prüfen.

Der öffentliche Schlüssel der Vertrauensstelle IRD für die Prüfung von Signaturen wird innerhalb der TI für die Gesundheitseinrichtungen an einem gesonderten Endpunkt zum regelmäßigen Download bereitgestellt.

Verschlüsselung von Daten

Das Verfahren zur Verschlüsselung der Daten sowie die konkreten Formate, die benutzt werden müssen, sind aktuell in Klärung.

Verschlüsselung der Eingangsdaten

Der Schutz der Datenübertragung durch die Gesundheitseinrichtung an die Vertrauensstelle IRD wird auf Protokollebene per TLS gesichert. Technisch bedingt sind für eine durchgehende Ende-zu-Ende-Verschlüsselung alle patienten- und/oder fall-identifizierenden Daten vor der Übertragung für die Vertrauensstelle auf Feldebene zu verschlüsseln.

Technisch wird dazu die jeweilige Klartextinformation an die entsprechende Schnittstelle des Konnektors bzw. des Basis-/KTR-Consumer gesendet werden. Nach der Operation der Verschlüsselung wird dann der verschlüsselte Wert zurückgegegben. Dieser Wert ist dann in der jeweiligen Eigenschaft anstelle der Klartextinformation des zu verwenden. Erweiterte Information zur Verwendung der Schnittstelle des Konnektors bzw. des Basis-/KTR-Consumer zur Verschlüsselung und zu den konkreten Formaten finden sich unter dem Punkt Übergreifene Beispiele.

Der öffentliche Schlüssel der Vertrauensstelle IRD für die Verschlüsselung wird innerhalb der TI für die Gesundheitseinrichtungen an einem gesonderten Endpunkt zum regelmäßigen Download bereitgestellt.

API

API für die Meldung einer Implantatmaßnahme (Erstmeldung, Storno, Korrektur,...) von Patienten mit einer deutschen Krankenversicherung

POST /psngen/api/v2.0/pseudonym/generate

Beschreibung

Diese Schnittstelle ist für die Meldung einer implantbezogenen Maßnahme zu einem Patienten mit einer deutschen Krankenversicherung zu benutzen.

Eingabedaten:

Die Daten, die von der API zu der Meldung einer implantatbezogene Maßnahme erwartet und verarbeitet werden, sind in der OpenApi-Spezifikation unter der Adresse https://xml.ir-d.de/rst/download/vst/v2.0/OpenApi-PseudonymGenerierung.json unter der Beschreibung des Typs "PseudonymisationRequest" beschrieben. Zu der Meldung von Daten zu einem Patienten mit deutscher Krankenversicherung werden in der Eigenschaft "Patientendaten" Daten vom Typ "PatientCaseData" erwartet.

Eigenschaft "Patientendaten"

- definiert die patienten- und fallidentifizierbaren Daten eines Patienten zu einer impalantatbezogene Maßnahme
- interner Typ der Eigenschaft: "PatientCaseData"
- Felder innerhalb der Eigenschaft "Patientendaten":
  - IdVersicherter
    - definiert den eindeutigen Identifizierer des Patienten. Dieser Identifizierer ist der unveränderbare Teil der Krankenversichertennummer (KVNR) nach § 290 des Fünften Buches Sozialgesetzbuch (SGB V) bzw. eine andere eindeutige, unveränderbare und nach einheitlichen Kriterien gebildete Identifikationsnummer entsprechend §17 Abs. 4 IRegG.
    - Optional: Nein
    - Format: Byte
    - Typ: String
    - Schutz: Der Wert muss für die Vertrauensstelle verschlüsselt werden
    - Prüfungen:
      - bei Angabe einer Krankenversichertennummer:
        - es wird geprüft, dass die Angabe einer gültigen KVNR entspricht (Regex: ^[A-Z]{1}[0-9]{9}$, sowie Prüfung der Prüfsumme)
        - es wird geprüft, dass die Angabe nicht die der definierten KVNR eines Patienten ohne deutsche Krankenversicherung ist ([1])
        - es wird geprüft, dass diese KVNR im jeweiligen Context benutzt werden darf ([2])
      - bei Angabe einer anderen eindeutigen, unveränderbare und nach einheitlichen Kriterien gebildete Identifikationsnummer:
        - es wird geprüft, ob die Angabe einen bekannten, festglegten Schema entspricht, sowie eine Prüfung der Prüfsumme
  - IdDatensatz
    - definiert die Datensatz-Id des betroffenen Falls
    - Optional: Nein
    - Format: Byte
    - Typ: String
    - Schutz: Der Wert muss für die Vertrauensstelle verschlüsselt werden
    - Prüfungen:
      - es wird geprüft, dass die Angabe einer gültigen Datensatz-Id entspricht (Regex: ^[a-zA-Z0-9.\-]{3,40}$)
      - es wird geprüft, dass dieser Datensatz-Identifikator im jeweiligen Context benutzt werden darf ([3])
  - IdKrankenversicherung
    - definiert die Id der Krankenversicherung (Haupt-IK) des Versicherten
    - Optional: Nein
    - Format: Byte
    - Typ: String
    - Schutz: Der Wert muss für die Vertrauensstelle verschlüsselt werden
    - Prüfungen:
      - es wird geprüft, dass diese Angabe einem gültigen Institutionskennzeichen (IK) entspricht (Regex: ^[0-9]{9}$, sowie die Prüfung der Prüfsumme)
      - es wird geprüft, dass die Angabe nicht die definierte IK der Krankenversicherung für einen Patienten ohne deutsche Krankenversicherung ist ([1])

Eigenschaft "Signatur"

- definiert die Signatur über die Patientendaten
- Optional: Nein
- Format: Byte
- Typ: String

Rückgabe

Nach erfolgreicher Verarbeitung der Daten wird der StatusCode 200 zurückgegeben. Als Content der Antwort wird die signierte Transfernummer zurückgegegben (Typ: "SignedTransferNumber").

Fehlerfälle

Fehler, die bei der Kommunikation mit der API oder bei der Verarbeitung von Daten auftreten können, werden der sendenden Gesundheitseinrichtung in Form von HTTP-Statuscodes zurück gemeldet.
Liste der Fehlerfälle:

- StatusCode 400: dieser Fehler wird zurückgemeldet, wenn die Nachricht fehlerhaft war, weil Pflichtangaben fehlen, oder Angaben nicht plausibel sind (siehe dazu Prüfungen von Feldern). Es werden in diesem Fehlerfall keine erweiterten Fehlerbeschreibungen zurückgemeldet.
- StatusCode 401: dieser Fehler wird zurückgemeldet, wenn die Authentifizierung an der Schnittstelle fehlgeschlagen ist.
- StatusCode 403: dieser Fehler wird zurückgemeldet, wenn die Anfrage nicht autorisiert war. Dies beinhaltet auch die Verwendung von produktiven Daten in der Referenz-Umgebung.
- StatusCode 415: dieser Fehler wird zurückgemeldet, wenn ein falscher Content-Type gesendet wurde.
- StatusCode 500: dieser Fehler wird zurückgemeldet, wenn ein interner Fehler bei der Verarbeitung aufgetreten ist.

Beispiel Request


```
{
   "Patientendaten": {
      "IdVersicherter":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZFZlcnNpY2hlcnRlcg==",
      "IdDatensatz":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZERhdGVuc2F0eg==",
      "IdKrankenversicherung":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlciBJZEtyYW5rZW52ZXJzaWNoZXJ1bmc="
   },
   "Signatur":"SWNoIGJpbiBkaWUgU2lnbmF0dXIgw7xiZXIgZGllIEVpbmdhbmdzZGF0ZW4="
}
```


Beispiel Response


```
{
   "Transfernummer": {
      "Wert":"a9bf75a7cf75a446ff9f1ca0c7ef06155cf4b356393de1d3f41309d8a2de31e4",
      "Signatur":"SWNoIGJpbiBkaWUgU2lnbmF0dXIgw7xiZXIgZGllIFRyYW5zZmVybnVtbWVy"
   }
}
```


Signaturbildung der Eingangsdaten

Die Signatur, die vom Sender an den Eingangsdaten mitzusenden ist, bezieht sich auf die (bereits verschlüsselten) Patientendaten. Bei der Signatur werden die Patientendaten in einer konkatenierten Form zur Signaturberechnung und -prüfung betrachtet.

Für beispielsweide einem JSON-Fragment mit diesen Daten


```
"Patientendaten": {
   "IdVersicherter":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZFZlcnNpY2hlcnRlcg==",
   "IdDatensatz":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZERhdGVuc2F0eg==",
   "IdKrankenversicherung":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlciBJZEtyYW5rZW52ZXJzaWNoZXJ1bmc="
}
```


wird folgende Zeichenkette erstellt:

```
<Wert des IdVersicherten> + | + <Wert des IdDatensatzes> + | + <Wert der IdKrankenversicherung>

Hexadezimale Darstellung
49 63 68 20 62 69 6E 20 65 69 6E 20 76 65 72 73 63 68 6C C3 BC 73 73 65 6C 74 65 72 20 57 65 72 74 20 64 65 73
20 49 64 56 65 72 73 69 63 68 65 72 74 65 72 7C 49 63 68 20 62 69 6E 20 65 69 6E 20 76 65 72 73 63 68 6C C3 BC
73 73 65 6C 74 65 72 20 57 65 72 74 20 64 65 73 20 49 64 44 61 74 65 6E 73 61 74 7A 7C 49 63 68 20 62 69 6E 20
65 69 6E 20 76 65 72 73 63 68 6C C3 BC 73 73 65 6C 74 65 72 20 57 65 72 74 20 64 65 72 20 49 64 4B 72 61 6E 6B
65 6E 76 65 72 73 69 63 68 65 72 75 6E 67
```

Weitere Informationen zu der Bildung der Signatur finden sich unter dem Punkt Signierung von Eingangsdaten.

Signierung der Ausgangsdaten

Bei erfolgreicher Verarbeitung der Anfrage wird der sendenden Gesundheitseinrichtung eine für die Registerstelle verschlüsselte und signierte Transfernummer zurückgegeben. Dieses Konstrukt "SignedTransferNumber" beinhaltet neben den Wert der Transfernummer auch die Signatur. Die Signatur wird dabei über den Wert der verschlüsselten Transfernummer gebildet.

Weitere Informationen finden sich unter dem Punkt Signierung der Ausgangsdaten.

API für die Meldung einer Implantatmaßnahme (Erstmeldung, Storno, Korrektur,...) von Patienten ohne eine deutsche Krankenversicherung

POST /psngen/api/v2.0/pseudonym/generate

Beschreibung

Diese Schnittstelle ist für die Meldung einer implantbezogenen Maßnahme zu einem Patienten ohne deutsche Krankenversicherung zu benutzen.

Eingabedaten:

Die Daten, die von der API zu der Meldung einer implantatbezogene Maßnahme erwartet und verarbeitet werden, sind in der OpenApi-Spezifikation unter der Adresse https://xml.ir-d.de/rst/download/vst/v2.0/OpenApi-PseudonymGenerierung.json unter der Beschreibung des Typs "PseudonymisationRequest" beschrieben. Zu der Meldung von Daten zu einem Patienten ohne deutsche Krankenversicherung werden in der Eigenschaft "Patientendaten" Daten vom Typ "PatientNonKvCaseData" erwartet.

Eigenschaft "Patientendaten"

- definiert die fallidentifizierbaren Daten eines Patienten ohne deutsche Krankenversicherung zu einer impalantatbezogene Maßnahme
- interner Typ der Eigenschaft: "PatientNonKvCaseData"
- Felder innerhalb der Eigenschaft "Patientendaten":
  - IstPatientOhneDtKv
    - definiert die eindeutige Angabe, dass es sich bei der Meldung um eine Meldung für einen Patienten ohne dt. Krankenversicherung handelt
    - Optional: Nein
    - Typ: String
    - Schutz: Der Wert wird in Plaintext übertragen
    - Prüfungen:
      - es wird geprüft, dass die Angabe dem festgelegten Wert entspricht (Regex: ^(?:true)$)
  - IdDatensatz
    - definiert die Datensatz-Id des betroffenen Falls
    - Optional: Nein
    - Format: Byte
    - Typ: String
    - Schutz: Der Wert muss für die Vertrauensstelle verschlüsselt werden
    - Prüfungen:
      - es wird geprüft, dass die Angabe einer gültigen Datensatz-Id entspricht (Regex: ^[a-zA-Z0-9.\-]{3,40}$)
      - es wird geprüft, dass dieser Datensatz-Identifikator im jeweiligen Context benutzt werden darf ([3])

Eigenschaft "Signatur"

- definiert die Signatur über die Patientendaten
- Optional: Nein
- Format: Byte
- Typ: String

Rückgabe

Nach erfolgreicher Verarbeitung der Daten wird der StatusCode 200 zurückgegeben. Als Content der Antwort wird die signierte Transfernummer zurückgegegben (Typ: "SignedTransferNumber").

Fehlerfälle

Fehler, die bei der Kommunikation mit der API oder bei der Verarbeitung von Daten auftreten können, werden der sendenden Gesundheitseinrichtung in Form von HTTP-Statuscodes zurück gemeldet.
Liste der Fehlerfälle:

- StatusCode 400: dieser Fehler wird zurückgemeldet, wenn die Nachricht fehlerhaft war, weil Pflichtangaben fehlen, oder Angaben nicht plausibel sind (siehe dazu Prüfungen von Feldern). Es werden in diesem Fehlerfall keine erweiterten Fehlerbeschreibungen zurückgemeldet.
- StatusCode 401: dieser Fehler wird zurückgemeldet, wenn die Authentifizierung an der Schnittstelle fehlgeschlagen ist.
- StatusCode 403: dieser Fehler wird zurückgemeldet, wenn die Anfrage nicht autorisiert war. Dies beinhaltet auch die Verwendung von produktiven Daten in der Referenz-Umgebung.
- StatusCode 415: dieser Fehler wird zurückgemeldet, wenn ein falscher Content-Type gesendet wurde.
- StatusCode 500: dieser Fehler wird zurückgemeldet, wenn ein interner Fehler bei der Verarbeitung aufgetreten ist.

Beispiel Request


```
{
   "Patientendaten": {
      "IstPatientOhneDtKv":"true",
      "IdDatensatz":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZERhdGVuc2F0eg=="
   },
   "Signatur":"SWNoIGJpbiBkaWUgU2lnbmF0dXIgw7xiZXIgZGllIEVpbmdhbmdzZGF0ZW4="
}
```


Beispiel Response

```
{
   "Transfernummer": {
      "Wert":"a9bf75a7cf75a446ff9f1ca0c7ef06155cf4b356393de1d3f41309d8a2de31e4",
      "Signatur":"SWNoIGJpbiBkaWUgU2lnbmF0dXIgw7xiZXIgZGllIFRyYW5zZmVybnVtbWVy"
   }
}
```

Signaturbildung der Eingangsdaten

Die Signatur, die vom Sender an den Eingangsdaten mitzusenden ist, bezieht sich auf die (bereits verschlüsselten) Patientendaten. Bei der Signatur werden die Patientendaten in einer konkatenierten Form zur Signaturberechnung und -prüfung betrachtet.

Für beispielsweide einem JSON-Fragment mit diesen Daten

```
"Patientendaten": {
   "IstPatientOhneDtKv":"true",
   "IdDatensatz":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZERhdGVuc2F0eg=="
}
```

wird folgende Zeichenkette erstellt:

```
<Wert der Eigenschaft IstPatientOhneDtKv> + | + <Wert des IdDatensatzes>

Hexadezimale Darstellung
74 72 75 65 7C 49 63 68 20 62 69 6E 20 65 69 6E 20 76 65 72 73 63 68 6C C3 BC 73 73 65 6C 74 65 72 20 57 65 72
74 20 64 65 73 20 49 64 44 61 74 65 6E 73 61 74 7A
```

Weitere Informationen zu der Bildung der Signatur finden sich unter dem Punkt Signierung von Eingangsdaten.

Signierung der Ausgangsdaten

Bei erfolgreicher Verarbeitung der Anfrage wird der sendenden Gesundheitseinrichtung eine für die Registerstelle verschlüsselte und signierte Transfernummer zurückgegeben. Dieses Konstrukt "SignedTransferNumber" beinhaltet neben den Wert der Transfernummer auch die Signatur. Die Signatur wird dabei über den Wert der verschlüsselten Transfernummer gebildet.

Weitere Informationen finden sich unter dem Punkt Signierung der Ausgangsdaten.

API für die Initiierung eines Auskunftsverlangens

POST /rfigen/api/v2.0/informationrequest/generate

Beschreibung

Diese Schnittstelle ist für die Initiierung eines Auskunftsverlangens eines Patienten entsprechend DSGVO zu benutzen. Durch die Initiierung des Auskunftsverlangens können Informationen zu den im Register gespeicherten Daten des Patienten angefragt werden.

Eingabedaten:

Die Daten, die von der API zu der Initiierung eines Auskunftsverlangens erwartet und verarbeitet werden, sind in der OpenApi-Spezifikation unter der Adresse https://xml.ir-d.de/rst/download/vst/v2.0/OpenApi-Auskunftsverlangen.json unter der Beschreibung des Typs "InformationRequestDataRequest" beschrieben.

Eigenschaft "Patientendaten"

- definiert die patientenidentifizierbaren Daten eines Patienten zu der Initiierung eines Auskunftsverlangens
- interner Typ der Eigenschaft: "InformationRequestData"
    - IdVersicherter
        - definiert den eindeutigen Identifizierer des Patienten. Dieser Identifizierer ist der unveränderbarer Teil der Krankenversichertennummer (KVNR) nach § 290 des Fünften Buches Sozialgesetzbuch (SGB V) bzw. eine andere eindeutige, unveränderbare und nach einheitlichen Kriterien gebildete Identifikationsnummer entsprechend §17 Abs. 4 IRegG.
        - Optional: Nein
        - Format: Byte
        - Typ: String
        - Schutz: Der Wert muss für die Vertrauensstelle verschlüsselt werden
        - Prüfungen:
            - bei Angabe einer Krankenversichertennummer:
                - es wird geprüft, dass die Angabe einer gültigen KVNR entspricht (Regex: ^[A-Z]{1}[0-9]{9}$, sowie Prüfung der Prüfsumme)

        - es wird geprüft, dass die Angabe nicht die der definierten KVNR eines Patienten ohne deutsche Krankenversicherung ist ([1])
        - es wird geprüft, dass diese KVNR im jeweiligen Context benutzt werden darf ([2])
    - bei Angabe einer anderen eindeutigen, unveränderbare und nach einheitlichen Kriterien gebildete Identifikationsnummer:
        - es wird geprüft, ob die Angabe einen bekannten, festglegten Schema entspricht, sowie eine Prüfung der PrüfsummeFelder innerhalb der Eigenschaft "Patientendaten":

Eigenschaft "Signatur"

- definiert die Signatur über die Patientendaten
- Optional: Nein
- Format: Byte
- Typ: String

Rückgabe

Nach erfolgreicher Verarbeitung der Daten wird der StatusCode 200 zurückgegeben. Als Content der Antwort wird die signierte Transfernummer sowie die Arbeitsnummer für die sendende Einrichtung zurückgegegben (Typ: "InformationRequestPersistenceResultResponse").

Fehlerfälle

Fehler, die bei der Kommunikation mit der API oder bei der Verarbeitung von Daten auftreten können, werden der sendenden Gesundheitseinrichtung in Form von HTTP-Statuscodes zurück gemeldet.
Liste der Fehlerfälle:

- StatusCode 400: dieser Fehler wird zurückgemeldet, wenn die Nachricht fehlerhaft war, weil Pflichtangaben fehlen, oder Angaben nicht plausibel sind (siehe dazu Prüfungen von Feldern). Es werden in diesem Fehlerfall keine erweiterten Fehlerbeschreibungen zurückgemeldet.
- StatusCode 401: dieser Fehler wird zurückgemeldet, wenn die Authentifizierung an der Schnittstelle fehlgeschlagen ist.
- StatusCode 403: dieser Fehler wird zurückgemeldet, wenn die Anfrage nicht autorisiert war. Dies beinhaltet auch die Verwendung von produktiven Daten in der Referenz-Umgebung.
- StatusCode 415: dieser Fehler wird zurückgemeldet, wenn ein falscher Content-Type gesendet wurde.
- StatusCode 500: dieser Fehler wird zurückgemeldet, wenn ein interner Fehler bei der Verarbeitung aufgetreten ist.

Beispiel Request


```
{
   "Patientendaten": {
      "IdVersicherter":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZFZlcnNpY2hlcnRlcg=="
   },
   "Signatur":"SWNoIGJpbiBkaWUgU2lnbmF0dXIgw7xiZXIgZGllIEVpbmdhbmdzZGF0ZW4="
}
```


Beispiel Response


```
{
   "Auskunftsverlangen": {
      "Transfernummer": {
         "Wert":"a9bf75a7cf75a446ff9f1ca0c7ef06155cf4b356393de1d3f41309d8a2de31e4",
         "Signatur":"SWNoIGJpbiBkaWUgU2lnbmF0dXIgw7xiZXIgZGFzIGdlc2FtdGUgRmVsZCAiUGF0aWVudGVuZGF0ZW4iLi4="
      },
      "Arbeitsnummer": "d85782a8756255afd857"
   },
   "Signatur":"SWNoIGJpbiBkaWUgU2lnbmF0dXIgw7xiZXIgZGFzIGdlc2FtdGUgRmVsZCAiUGF0aWVudGVuZGF0ZW4iLi4="
}
```


Signaturbildung der Eingangsdaten

Die Signatur, die vom Sender an den Eingangsdaten mitzusenden ist, bezieht sich auf die (bereits verschlüsselten) Patientendaten. Bei der Signatur werden die Patientendaten in einer konkatenierten Form zur Signaturberechnung und -prüfung betrachtet.

Für beispielsweide einem JSON-Fragment mit diesen Daten


```
"Patientendaten": {
   "IdVersicherter":"SWNoIGJpbiBlaW4gdmVyc2NobMO8c3NlbHRlciBXZXJ0IGRlcyBJZFZlcnNpY2hlcnRlcg=="
}
```

wird folgende Zeichenkette erstellt:

```
<Wert der Eigenschaft IdVersicherter>

Hexadezimale Darstellung
49 63 68 20 62 69 6E 20 65 69 6E 20 76 65 72 73 63 68 6C C3 BC 73 73 65 6C 74 65 72 20 57 65 72 74 20 64 65 73
20 49 64 56 65 72 73 69 63 68 65 72 74 65 72
```

Weitere Informationen zu der Bildung der Signatur finden sich unter dem Punkt Signierung von Eingangsdaten.

Signierung der Ausgangsdaten

Bei erfolgreicher Verarbeitung der Anfrage wird der sendenden Gesundheitseinrichtung zum einen eine für die Registerstelle verschlüsselte und signierte Transfernummer sowie auch die Arbeitsnummer für dieses Auskunftsverlangen zurückgegeben.

Das Konstrukt "SignedTransferNumber" beinhaltet neben den Wert der Transfernummer auch die Signatur. Die Signatur wird dabei über den Wert der verschlüsselten Transfernummer gebildet.

Über die gesamte Antwort (inklusive der Arbeitsnummer) wird auch zusätzlich eine Signatur gebildet. Dieses Vorgehen ist so gewählt, damit zum einen die Registerstelle die Signatur der Transfernummer prüfen kann, und die Gesundheitseinrichtung die Signatur der gesamten Nachricht.

Für beispielsweide einem JSON-Fragment mit diesen Daten

```
{
   "Auskunftsverlangen": {
      "Transfernummer": {
         "Wert":"a9bf75a7cf75a446ff9f1ca0c7ef06155cf4b356393de1d3f41309d8a2de31e4",
         "Signatur":"SWNoIGJpbiBkaWUgU2lnbmF0dXIgw7xiZXIgZGllIFRyYW5zZmVybnVtbWVy"
      },
      "Arbeitsnummer": "d85782a8756255afd857"
   },
   "Signatur":"SWNoIGJpbiBkaWUgU2lnbmF0dXIgw7xiZXIgZGllIHp1csO8Y2tnZWdlYmVuZW4gRGF0ZW4="
}
```

wird folgende Zeichenkette erstellt:

```
<Wert der Transfernummer> + | + <Signatur der Transfernummer> | + <Wert der Arbeitsnummer>

Hexadezimale Darstellung
61 39 62 66 37 35 61 37 63 66 37 35 61 34 34 36 66 66 39 66 31 63 61 30 63 37 65 66 30 36 31 35 35 63 66 34 62
33 35 36 33 39 33 64 65 31 64 33 66 34 31 33 30 39 64 38 61 32 64 65 33 31 65 34 7C 49 63 68 20 62 69 6E 20 64
69 65 20 53 69 67 6E 61 74 75 72 20 C3 BC 62 65 72 20 64 61 73 20 67 65 73 61 6D 74 65 20 46 65 6C 64 20 22 50
61 74 69 65 6E 74 65 6E 64 61 74 65 6E 22 2E 2E 7C D8 57 82 A8 75 62 55 AF D8 57
```

Weitere Informationen finden sich unter dem Punkt Signierung der Ausgangsdaten.

Übergreifende Beispiele

> *die hier folgenden Beispiele zu Verschlüsselung und Signierung der Daten sind derzeit noch in Klärung*

Nichtfunktionale Festlegungen

aktuell in Klärung

Test

Definition von Testdaten

Die Regeln für die Testdaten stellen sicher, dass patienten-identifizierende Daten nicht in Testumgebungen (für IRD die Referenzumgebung der TI) verwendet werden. Hierbei sind die speziellen Regeln für die Referenzumgebung der TI und die Produktivumgebung der TI zu beachten[2] [3].

Testdaten Krankenversichertennummer (KVNR)

Der GKV-SV hat einen KVNR-Nummernkreis bereitgestellt, der ausschließlich für Tests im Rahmen des Implantateregisters Deutschland (IRD) genutzt werden darf. Die KVNRs dieses Nummernkreises werden gesichert keiner Person zugeordnet.

Definition: Nummernkreis A1111xxxxP , wobei x für eine Ziffer von 0 bis 9 und P für die Prüfziffer steht.

Testdaten Datensatz-Identifikator (IdDatensatz)

Für Tests ist als Datensatz-Identifikator (IdDatensatz) eine Zeichenfolge zu verwenden, die nach den folgenden Regeln aufgebaut ist:

```
Regex: ^TESTONLY[a-zA-Z0-9.\-]{3,32}$
```

Beispiel: "TESTONLY1234"

Fußnoten

1.) Die Angabe der speziellen Krankenversichertennummer (KVNR) und des speziellen Institutionskennzeichens der Krankenversicherung bei Meldungen für Patienten ohne deutsche Krankenversicherung der API Version 1.1 werden in dieser API-Version nicht mehr akzeptiert. Für Implantatmeldungen für diese Patientengruppe steht jetzt ein separates API zur Verfügung.

2.) Für die Verwendung von KVNR gelten folgende Einschränkungen:
Referenzumgebung der TI: In dieser Umgebung dürfen nur vordefinierte Nummern aus dem Testnummernkreis (Definition Testdaten) verwendet werden. Bei der Verwendung von prod. KVNR-Nummern in der Referenz-Umgebung wird die Meldung nicht angenommen und dem Sender ein Fehler zuückgegeben.

Produktivumgebung der TI: In dieser Umgebung werden nur produktive Daten erwartet. Eine Verwendung einer produktiven KVNR, d.h. einer potentiell einem Versicherten zugeordneten KVNR, als Testdatensatz ist in dieser Umgebung verboten. Als Ausnahme sind Tests der korrekten Anbindung der meldenden Gesundheitseinrichtung an die Produktivumgebung der TI ausschließlich mit einer KVNR aus dem Testnummernkreis (Definition Testdaten) zugelassen.

3.) Für die Verwendung von Datensatz-Identifikatoren gelten folgende Einschränkungen:
Referenzumgebung der TI: In dieser Umgebung dürfen nur Datensatz-Identifikatoren verwendet werden, die dem Testdatenschema entsprechen (Definition Testdaten). Bei der Verwendung von produktiven Datensatz-Identifikatoren in der Referenzumgebung wird die Meldung nicht angenommen und dem Sender ein Fehler zuückgegeben.

Produktivumgebung der TI: In dieser Umgebung werden nur produktive Daten erwartet. Eine Verwendung eines produktiven Datensatz-Identifikators als Testdatensatz ist in dieser Umgebung verboten. Als Ausnahme für Tests der Anbindung an dieser Umgebung sind nur Datensatz-Identifikatoren zugelassen, die dem Testdatenschema (Definition Testdaten) entsprechen.

Kontakt

Robert Koch-Institut
Nordufer 20
13353 Berlin

E-Mail: Vertrauensstelle-IRD@rki.de