Institut für Qualitätssicherung und
Transparenz im Gesundheitswesen

Systemweit einheitliche Leistungserbringer-pseudonymisierung

Anwenderhandbuch Pseudonymisierungsprogramm V05

Erstellt im Auftrag des
Gemeinsamen Bundesausschusses

Stand: 20. November 2019

Impressum

Thema:

Pseudonymisierungsprogramm zur systemweit einheitlichen Leistungserbringerpseudonymisierung. Anwenderhandbuch V05

Auftraggeber:

Gemeinsamer Bundesausschuss

Datum der Abgabe:

20. November 2019

Herausgeber:

IQTIG – Institut für Qualitätssicherung
und Transparenz im Gesundheitswesen

Katharina-Heinroth-Ufer 1
10787 Berlin

Telefon: (030) 58 58 26-0
Telefax: (030) 58 58 26-999

info@iqtig.org
http://www.iqtig.org

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Informationen zu diesem Dokument

Darstellungsmittel

Im Folgenden sind Symbole und Darstellung besonderer Informationen beschrieben.

Achtung

Beschreibt Ursache, Folge und Vermeidung einer besonderen Fehlanwendung, die zu Problemen bei der Implementierung oder Ähnlichem führen kann.

Beispiel:

Beispiele sind ein Hilfsmittel, um zuvor vermittelte Informationen oder konkrete Abschnitte der Anwendung zu verdeutlichen.

Zielgruppe

Dieses Handbuch richtet sich an administrative Mitarbeiter der Datenannahmestellen (DAS) und Softwareentwickler, die mit der Umsetzung der Leistungserbringerpseudonymisierung beschäftigt sind.

Änderungen in der Version 02 (gegenüber Version 2015 V1.0)

- Anpassung des Dokuments an das Layout des Instituts nach § 137a SGB V (IQTIG)
- Redaktionelle Änderungen

Änderungen in der Version 04 (gegenüber Version 03)

- Anpassung der Tabelle mit den gültigen Datenannahmestellen

Änderungen in der Version 05 (gegenüber Version 04)

- Erweiterung um die Pseudonymisierungsmöglichkeit der Standort-ID
- Entfernen der „Übertragung der Pseudonyme an die BAS“ (veraltet; Neukonzeption in Arbeit)
- Detaillierte Darstellung s. Kapitel 5 Änderungen der Version 05 / PSP 2.0.15 (Details)

1 Einleitung

Die durch die Richtlinie zur einrichtungs- und sektorenübergreifenden Qualitätssicherung (Qesü-RL)[1] vorgenommene Ausdehnung der Qualitätssicherung vom stationären auf den vertragsärztlichen Bereich sowie die Einbeziehung von Sozialdaten bei den Krankenkassen in die Qualitätsauswertungen erfordern eine einheitliche Pseudonymisierung der leistungserbringeridentifizierenden Daten. Die systemweit einheitliche Pseudonymisierung bildet die Grundlage für die Zusammenführung der Qualitätssicherungsdaten (QS-Daten) aus den unterschiedlichen Datenquellen in der Bundesauswertungsstelle (BAS) zur Berechnung der Qualitätsindikatoren sowie für den Versand der Rückmeldeberichte von der BAS an die Leistungserbringer (LE). Die Qesü-RL fordert die systemweit einheitliche Leistungserbringerpseudonymisierung (LE-Pseudonymisierung) in § 3 Abs. 2 Satz 4 der Anlage zu Teil 1 der Richtlinie:

> *Für die Pseudonymisierung stimmen die Datenannahmestellen nach § 9 Absatz 1 Satz 2 der Richtlinie (KV[2] bzw. KZV[3]), die Datenannahmestellen nach § 9 Absatz 1 Satz 3 der Richtlinie (LQS[4]/LKG[5]) sowie die Datenannahmestelle nach § 9 Absatz 1 Satz 5 der Richtlinie (DAS-KK[6]) untereinander ein Verfahren ab, welches sicherstellt, dass die Datenannahmestellen den gleichen leistungserbringeridentifizierenden Daten jeweils das gleiche Pseudonym zuordnen.*

Die Datenflussspezifikation zur systemweit einheitlichen Leistungserbringerpseudonymisierung beschreibt die Datenflüsse, Dateiformate zum Datenaustausch sowie algorithmische Grundlagen der LE-Pseudonymisierung. Das Dokument richtet sich an die o. g. Datenannahmestellen und an die mit der Umsetzung beauftragten Softwarehersteller.

Die Erzeugung von kryptografischen Schlüsseln und von Zertifikaten und die Pseudonymisierung sowie Depseudonymisierung von leistungserbringeridentifizierenden Daten werden mithilfe des Leistungserbringerpseudonymisierungsprogramms durchgeführt. Die Anwenderdokumentation für das Programm befindet sich in der Datei „Pseudonymisierungsprogramm.pdf“ und ergänzt die Datenflussspezifikation. Auf der Website http://www.iqtig.org kann die aktuelle Version des Programms, der Anwenderdokumentation und der Datenflussspezifikation als ZIP-Archiv heruntergeladen werden.

1.1 Verschlüsselungsverfahren zur Pseudonymisierung

Die Erstellung der LE-Pseudonyme wird mittels eines asymmetrischen (Public-Key-)Verschlüsselungsverfahrens durchgeführt. Im Gegensatz zu symmetrischen Verschlüsselungsverfahren wird kein gemeinsamer geheimer Schlüssel benötigt, den beide Kommunikationspartner kennen. Bei

[1] https://www.g-ba.de/informationen/richtlinien/72/
[2] Kassenärztliche Vereinigung (KV)
[3] Kassenzahnärztliche Vereinigung (KZV)
[4] Landesgeschäftsstelle für Qualitätssicherung (LQS)
[5] Landeskrankenhausgesellschaft (LKG)
[6] Datenannahmestelle für die Krankenkassen (DAS-KK)

der asymmetrischen Verschlüsselung wird vom Benutzer (LQS/LKG/KV/KZV) ein Schlüsselpaar, bestehend aus einem privaten Schlüssel und einem öffentlichen Schlüssel, generiert. Die Schlüssel sind gemeinsam mit weiteren Informationen in Zertifikate eingebettet. Der private Schlüssel wird bei der Erzeugung des Schlüsselpaars mit einem Passwort geschützt. Der öffentliche Schlüssel wird an alle Kommunikationspartner verteilt. Mithilfe des öffentlichen Schlüssels können nun Daten verschlüsselt und sicher an den Besitzer des privaten Schlüssels übermittelt werden. Mit dem privaten Schlüssel und dem Passwort können die empfangenen Daten entschlüsselt werden. Die öffentlichen Schlüssel sind in Zertifikate eingebettet, die zusätzliche Metadaten enthalten.

Für die LE-Pseudonymisierung wird ein deterministisches, asymmetrisches Verschlüsselungsverfahren gebraucht. Verwendung findet der RSA-Verschlüsselungsalgorithmus von *Legion of the Bouncy Castle Inc*[7]*.* Dieser Algorithmus erfüllt sowohl die Anforderungen an die systemweit einheitliche Pseudonymisierung als auch die vom Bundesamt für Sicherheit in der Informationstechnik (BSI) empfohlenen Kriterien. Die deterministische Verschlüsselung nimmt in diesem Zusammenhang einen besonderen Stellenwert ein, da sichergestellt ist, dass die Verschlüsselung derselben LE-identifizierenden Daten mit dem gleichen öffentlichen Schlüssel bei verschiedenen DAS immer das gleiche Ergebnis produziert.

Achtung

Die Zertifikate mit den öffentlichen Schlüsseln dürfen nur den im Datenfluss vorgesehenen Stellen zur Verfügung gestellt werden. Es ist z. B. falsch, diese Zertifikate auf einer Website oder mit der Spezifikation zu veröffentlichen. In diesem Fall würde durch die Nutzung der deterministischen Verschlüsselung ein Besitzer von IKNR/BSNR die jeweiligen Pseudonyme erstellen können. Das Zertifikat, welches den privaten Schlüssel enthält, dient nur dazu, Rückmeldeberichte mit Sozialdaten dem richtigen Leistungserbringer zuordnen zu können, und darf daher Dritten nicht zur Verfügung gestellt werden.

1.2 Anwendungsbereich

Das Pseudonymisierungsprogramm wird für unterschiedliche Zielgruppen und Anwendungsmöglichkeiten in folgenden Varianten bereitgestellt:

- mit einer grafischen Benutzeroberfläche (siehe Abschnitt 3.1)
- als Konsolenanwendung (siehe Abschnitt 3.2)
- als Programmierschnittstelle (siehe Abschnitt 2.3)

Die grafische Benutzeroberfläche wurde für Testzwecke und die manuelle Integration der Leistungserbringerpseudonymisierung in den Datenfluss einer Datenannahmestelle erstellt. Eine Form der manuellen Integration wäre z. B., wenn eine DAS alle bekannten IKNR/BSNR mithilfe der Benutzeroberfläche pseudonymisiert, eine Mappingtabelle zwischen Pseudonym und Klartext erstellt und diese dann zu Ver- und Entschlüsselung einsetzt.

[7] https://www.bouncycastle.org/

Die Konsolenanwendung wurde für die programmatische Integration der Leistungserbringerpseudonymisierung in nicht Java-basierte Datenflüsse bei den Datenannahmestellen erstellt.

Zusätzlich steht den Java-Entwicklern bei den Datenannahmestellen eine API (Application Programming Interface/Programmierschnittstelle) zur Verfügung.

Zusatzinformationen zum Leistungserbringer wie Standorte oder Fachabteilungen können mit dem Tool über eine Webservice-Schnittstelle (SOAP) in Form von CSV-Dateien an die Bundesauswertungsstelle übertragen werden (siehe Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.**).

1.3 Eigenschaften der LE-Pseudonymisierung

Das Pseudonymisierungsprogramm nutzt einen deterministischen, asymmetrischen RSA-Verschlüsselungsalgorithmus mit einer Schlüssellänge von 2048 Bit zur Erzeugung von Pseudonymen. Die resultierenden Binärdaten werden dann in einen Base64-kodierten Text überführt und als Pseudonym benutzt.

Ein IKNR/BSNR-Pseudonym ist folgendermaßen aufgebaut:

- Art der Datenannahme (BAS/LQS/KV/SV)
- Bundesland/Region
- spezifizierte Trennzeichen ($$, ##)
- Mit dem öffentlichen Schlüssel verschlüsselte IKNR/BSNR

Beispiel:
LQS$$HH##fL1PqnoBllAaM2bXSXvf1MOnhp4NhPx2sKqnlng

Ein StandortID-Pseudonym ist folgendermaßen aufgebaut:

- Art der Datenannahme (BAS/LQS/KV/SV)
- Bundesland/Region
- spezifizierte Trennzeichen (**, ##)
- Mit dem öffentlichen Schlüssel verschlüsselte IKNR/BSNR

Beispiel:
LQS**HH##fL1PqnoBllAaM2bXSXvf1MOnhp4NhPx2sKqnlng

Die einheitliche Pseudonymisierung ist von allen Datenannahmestellen gemäß Qesü-RL durchzuführen. Für die Krankenhäuser ist die Institutskennzeichen-Nummer (IKNR) und die Standort-ID[8] zu pseudonymisieren, im kollektiv- bzw. selektivvertraglichen Bereich ist stattdessen die Betriebsstättennummer (BSNR) (XML-Element `BSNRAMBU`) zu pseudonymisieren. Die DAS für die

[8] Offizielle Standort-ID des bundesweiten Verzeichnis der Standorte der nach § 108 SGB V zugelassenen Krankenhäuser und ihrer Ambulanzen

Krankenkassen und die DAS für die selektivvertraglichen Arztpraxen (DAS-SV) besitzen keine eigenen Zertifikate und müssen deshalb die öffentlichen Zertifikate der DAS-LE auf Landesebene nutzen. Dazu müssen diese DAS entscheiden, ob es sich bei den leistungserbringeridentifizierenden Daten um eine IKNR und Standort-ID oder BSNR handelt. Die IKNRs und Standort-ID werden mit dem Schlüssel der LQS verschlüsselt, die BSNRs mit dem Schlüssel der KV. Zusätzlich muss auch das Bundesland bekannt sein, damit der richtige Schlüssel auf Landesebene ausgewählt werden kann.

Während der Durchführung der Pseudonymisierung wird das QS-Verfahren dem Pseudonymisierungsprogramm als Parameter übergeben. Dies geschieht beispielsweise in der CSV-Datei, die auf der grafischen Oberfläche als Eingabedatei angegeben werden kann. Das Pseudonymisierungsprogramm enthält in der jeweils aktuellsten Version alle gültigen QS-Verfahren, bei denen die systemweit einheitliche LE-Pseudonymisierung angewendet wird und prüft die Eingabedaten auf ihre Korrektheit. Somit wird verhindert, dass durch fehlerhafte Angaben des QS-Verfahrens in einer DAS ungültige LE-Pseudonyme erstellt werden.

2 Installation

2.1 Systemvoraussetzungen

Hardware

- mindestens 128 MB Arbeitsspeicher (RAM)
- mindestens 25 MB freier Festplattenspeicher

Software

- Betriebssystem, welches die Installation und Ausführung von Oracle Java ermöglicht
- Oracle Java 8
  (siehe auch https://www.java.com/de/download/help/sysreq.xml)

2.2 Programmbestandteile

Das Pseudonymisierungsprogramm mit der grafischen Benutzeroberfläche besteht aus zwei Dateien:

- Ausführbares Java-Programm
- Eigenschaftsdatei im Benutzer-Heim-Verzeichnis

Weitere Dateien sind für das Pseudonymisierungsprogramm bei der Datensicherung oder Datenreorganisation nicht zu berücksichtigen.

2.3 Referenzimplementierung

Zur Vereinfachung der Einbindung der API in Softwareentwicklungsprojekte wurde diese mit ihrer Referenzimplementierung als Maven-Artefakt bereitgestellt, sodass sie in das entsprechende lokal verwendete Repository deployed werden kann. Die Referenzimplementierung des Pseudonymisierungsprogrammes befindet sich in `pseudonymisierung-impl.jar`.

Sollte eine Datenannahmestelle die API verwenden wollen, ohne Maven einzusetzen, so finden sich die Informationen über abhängige Bibliotheken in den pom.xml-Dateien.

Alle benötigten Dateien befinden sich im Verzeichnis
`Programm\API_und_Referenzimplementierung`
innerhalb des PSP-ZIP-Archivs.

2.4 Installation durchführen

Java

Das Pseudonymisierungsprogramm wurde in der Programmiersprache Java entwickelt. Zur Ausführung von Java-Programmen wird das Java Runtime Environment (JRE) zur Verfügung gestellt. Das JRE kann „online“ und „offline“ installiert werden.

Offizielle Installationspakete können von der Website
http://www.java.com/de/download/manual.jsp

heruntergeladen werden.

Anleitungen zur Installation von Java unter verschiedenen Betriebssystemen sind auf der folgenden Website detailliert beschrieben:
https://www.java.com/de/download/help/download_options.xml

Pseudonymisierungsprogramm
Die grafische Benutzeroberfläche für das Pseudonymisierungsprogramm ist ein ausführbares Java-Programm und kann bei korrekt installierter Java-Laufzeitumgebung einfach mit einem Doppelklick unter Windows-Systemen gestartet werden.

Das Konsolenprogramm kann über die Windows-Kommandozeile aufgerufen werden.

2.5 Anpassung der Konfigurationsdatei

Die grafische Benutzeroberfläche speichert die Konfiguration des Programms in einer Eigenschaftsdatei (Properties). Diese wird im Benutzer-Heim-Verzeichnis angelegt.

Zum Beispiel unter Windows 7:
`C:\Users\M.Mustermann\pseudonym-config.properties`

Beim ersten Start des Programms wird diese Datei aus den mitgelieferten Initialeinstellungen befüllt und kann per Texteditor oder auf der Konfigurationsseite der grafischen Benutzeroberfläche angepasst werden.

2.6 Programm-Updates

Das Pseudonymisierungsprogramm wird in unregelmäßigen Intervallen Updates erhalten. Gründe für Updates können Fehlerbehebungen, Programmverbesserungen sowie die Anpassung des Programms an neue oder geänderte QS-Verfahren sein. Die alte Programmversion kann durch die neue Version ersetzt werden, indem die JAR-Datei ausgetauscht wird. Es besteht auch die Möglichkeit, mehrere Programmversionen parallel zu nutzen. Jede Programmversion nutzt die gemeinsame Konfigurationsdatei im Benutzer-Heim-Verzeichnis, sodass einmal vorgenommene Einstellungen nicht verloren gehen oder erneut durchgeführt werden müssen.

3 Programmaufbau

Das Pseudonymisierungsprogramm wird in unterschiedlichen Varianten bereitgestellt, deren Funktionsweise in den nächsten Unterkapiteln genauer erläutert wird.

3.1 Grafische Benutzeroberfläche

Datenannahmestellen können die grafische, interaktive Benutzeroberfläche nutzen, um das Pseudonymisierungsprogramm zu konfigurieren und Pseudonymisierungs- bzw. Depseudonymisierungsprozesse zu starten. Mit dem Programm können zudem weitere Zusatzinformationen an die Bundesauswertungsstelle (BAS) übermittelt werden.

3.1.1 Vorkonfiguration des Programms

Bei der erstmaligen Verwendung des Pseudonymisierungsprogramms ist es zur Verringerung redundanter Datenübergaben sinnvoll, die wiederkehrenden Dateneingaben vorab zu konfigurieren. Eine Überschreibung der Standardkonfiguration ist bei jedem Anwendungsfall möglich.

Beim ersten Start wird die Konfigurationsdatei `pseudonym-config.properties` im Benutzer-Heim-Verzeichnis abgelegt. Die hierbei angegebenen Spalten und Zeilennummern beginnen mit Spalte 0 und Zeile 1.

Tabelle 1: Eingabe- und Auswahlfelder der Pseudonymisierungs-, Depseudonymisierungs- und Konfigurationsoberfläche

Feld	Beschreibung
CSV-Datei-Spalte mit Eingangswert (Leistungserbringer)	***Pseudonymisierung:*** In dieser Spalte steht der zu pseudonymisierende Klartext (IKNR/BSNR). ***Depseudonymisierung:*** In dieser Spalte steht das Leistungserbringer-Pseudonym.
CSV-Datei-Spalte mit Ergebniswert (Leistungserbringer)	Die Spalte der CSV-Datei, in die der Ergebniswert des Leistungserbringer-Pseudonymisierungsvorgangs eingetragen wird. Ist der Wert größer als die Anzahl der Spalten in der CSV-Datei, wird das Ergebnis als letzte Spalte in die CSV-Datei eingetragen. Ist der Wert kleiner als 0, wird der Eingangswert mit dem Ergebnis überschrieben.
CSV-Datei-Spalte mit Eingangswert (Standort)	***Pseudonymisierung:*** In dieser Spalte steht der zu pseudonymisierende Klartext (STO-ID). ***Depseudonymisierung:*** In dieser Spalte steht das Standort-Pseudonym. *Die Angabe kann auch leer gelassen werden.*
CSV-Datei-Spalte mit Ergebniswert (Standort)	Die Spalte der CSV-Datei, in die der Ergebniswert des Standort-Pseudonymisierungsvorgangs eingetragen wird.

Feld	Beschreibung
	Ist der Wert größer als die Anzahl der Spalten in der CSV-Datei, wird das Ergebnis als letzte Spalte in die CSV-Datei eingetragen. Ist der Wert kleiner als 0, wird der Eingangswert mit dem Ergebnis überschrieben. *Die Angabe kann auch leer gelassen werden.*
CSV-Datei-Spalte mit Pseudonymisierungsverfahren	Gibt an, in welcher Spalte der CSV-Datei die Pseudonymisierungsmethode angegeben ist. Wird dieses Feld auf einen Wert kleiner 0 gestellt, so wird der Wert des Felds bzw. Parameters „Pseudonymisierungsverfahren“ für die gesamte CSV-Datei angewendet.
Datenannahmestellentyp	Typ der Datenannahmestelle auswählen (LQS, KV, BAS)
Eingangs-CSV-Dateiname	Die CSV-Datei, welche die Eingangsdaten enthält.
Datum Gültigkeitsende des Schlüssels	Gibt an, bis zu welchem Datum das nächste erzeugte Schlüsselpaar gültig ist. Bleibt dieses Feld leer, ist das Schlüsselpaar ab der Erzeugung 20 Jahre lang gültig.
Ergebnis-CSV-Dateiname	Die CSV-Datei, in welcher das Pseudonymisierungsprogramm das pseudonymisierte bzw. depseudonymisierte Ergebnis ablegen soll.
Landeskennung	Auswahl der Landeskennung.
Passwort, mit dem der private Schlüssel geschützt wird	Der private Schlüssel wird verschlüsselt in einem X.509-Zertifikat abgelegt. Das Passwort wird benötigt, um den privaten Schlüssel benutzen zu können.
QS-Verfahren	Das QS-Verfahren, auf welches sich die Pseudonymisierung bezieht. Der Wert des Dropdown-Menüs wird herangezogen, wenn bei „CSV-Datei-Spalte mit Pseudonymisierungsverfahren“ der Wert kleiner 0 ist.
Schlüsselverzeichnis	Das Verzeichnis, in dem die kryptografischen Schlüssel abgelegt sind.
Datum Gültigkeitsstart des Schlüssels	Gibt an, ab welchem Datum das nächste erzeugte Schlüsselpaar gültig ist.
Zeilennummer der CSV-Datei, mit der die Pseudonymisierung beginnen soll	Für den Fall, dass nicht die gesamte CSV-Datei pseudonymisiert werden soll, oder wenn es eine Kopfzeile mit Spaltennamen in der CSV-Datei gibt, kann angegeben werden, ab welcher Zeile die CSV-Datei bearbeitet werden soll.
Zusatzdaten CSV-Dateiname	Die CSV-Datei mit den Zusatzinformationen (Standortnummern, Haupt-IKNR/BSNR, Gültigkeit)

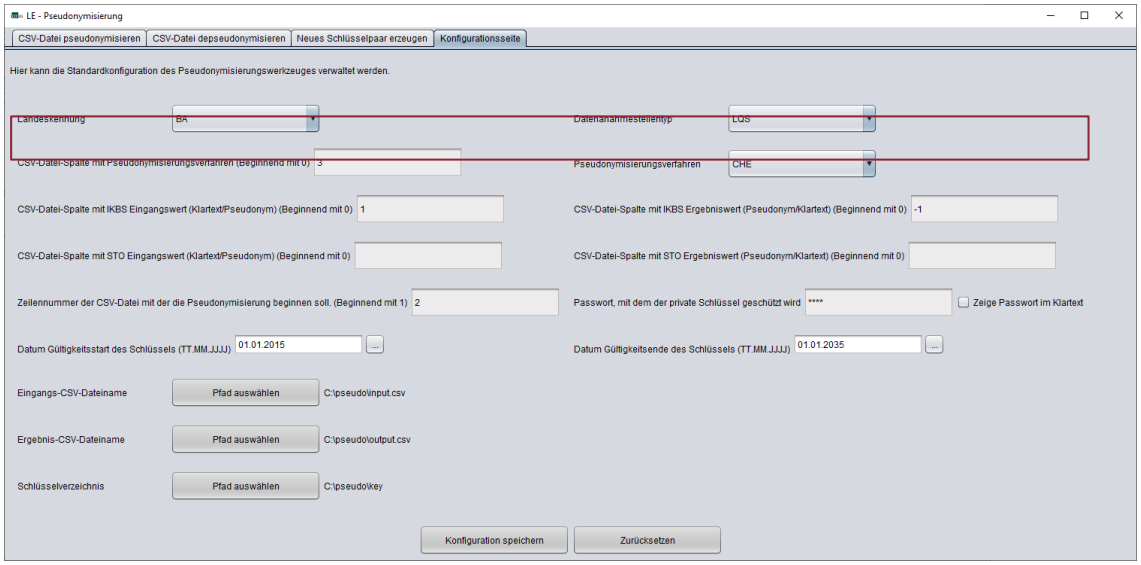

Abbildung 1: Konfigurationsoberfläche des LE-Pseudonymisierungsprogrammes

3.1.2 Schlüsselpaargenerierung

Bei der Generierung der Schlüsselpaare (X.509-Zertifikate) sind folgende Hinweise zu berücksichtigen:

- Nur die Datenannahmestellen der Krankenhäuser (LQS/LKG) und die Kassenärztlichen Vereinigungen (KV) bzw. Kassenzahnärztlichen Vereinigungen (KZV) dürfen öffentliche und private Zertifikate generieren
- Die DAS-KK und DAS-SV erhält für den Regelbetrieb von den LQS/LKG/KV/KZV über einen sicheren Kanal die öffentlichen Zertifikate
- Private Zertifikate dürfen nur vom Zertifikatsbesitzer selbst für die Depseudonymisierung verwendet werden
- Die Datenflüsse zur Übermittlung der Zertifikate zwischen den einzelnen Stellen werden in der Datenflussspezifikation beschrieben

Während der Generierung der Zertifikate werden verschiedene Attribute abgefragt (siehe Tabelle 2).

Tabelle 2: Eingabe- und Auswahlfelder der Oberfläche zur Schlüsselpaarerzeugung

Feld	Beschreibung
Datenannahmestellentyp	Art der Datenannahmestelle auswählen (LQS, KV, BAS).
Landeskennung	Auswahl der Landeskennung.
Startgültigkeitsdatum des Schlüssels	Gibt an, ab welchem Datum das nächste erzeugte Schlüsselpaar gültig ist.

Feld	Beschreibung
Endgültigkeitsdatum des Schlüssels	Gibt an, bis zu welchem Datum das nächste erzeugte Schlüsselpaar gültig ist. Bleibt dieses Feld leer, so bleibt das Schlüsselpaar ab der Erzeugung 20 Jahre lang gültig. Wir empfehlen, den Gültigkeitszeitraum nicht zu kurz zu wählen, da sich mit jedem neuen Schlüssel die Pseudonyme ändern und erneut Listen von zusammengehörigen Pseudonymen an die BAS übertragen werden müssen.
Passwort, mit dem der private Schlüssel geschützt wird	Der private Schlüssel wird verschlüsselt in einem X.509-Zertifikat abgelegt. Das Passwort wird benutzt, um den privaten Schlüssel benutzen zu können. Sollte dieses Passwort verloren gehen, müssen die Daten des laufenden Erfassungsjahres repseudonymisiert und erneut an die BAS übertragen werden.
Schlüssel Verzeichnis	Das Verzeichnis, in dem die Schlüssel abgelegt werden.

Die Ausgabe besteht aus zwei Dateien, dem öffentlichen und dem privaten Zertifikat. Die Dateien werden automatisch nach folgendem Schema benannt:

- Das öffentliche Zertifikat:
  `<Art-Datenannahme>$$<Bundeslandkürzel><Startgültigkeitsdatum>.cer`
- Das private Zertifikat:
  `<Art-Datenannahme>$$<Bundeslandkürzel><Startgültigkeitsdatum>.p12`

Beispiel (LQS Hamburg):
Öffentliches Zertifikat: `LQS$$HH20150501.cer`
Privates Zertifikat: `LQS$$HH20150501.p12`

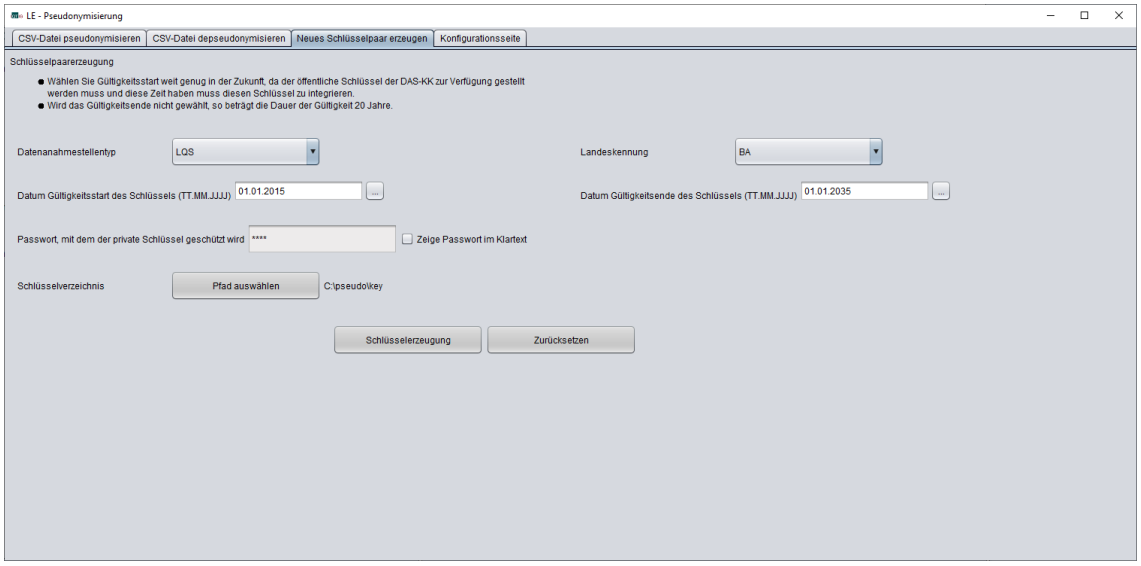

Abbildung 2: Oberfläche zur Generierung der Schlüsselpaare

Beim Klick auf die Schaltfläche „Zurücksetzen“ werden die Formularwerte auf die Werte der Konfigurationsseite zurückgesetzt.

3.1.3 Durchführung der Pseudonymisierung

Das Pseudonymisierungsprogramm nutzt den in Abschnitt 1.3 beschriebenen Algorithmus zur Durchführung der LE-Pseudonymisierung. Die Oberfläche zur Pseudonymisierung ist in Abbildung 3 dargestellt. Eine detaillierte Beschreibung der entsprechenden Auswahlfelder befindet sich in Tabelle 1.

Weitere Informationen über die LE-Daten und den Aufbau der Exportdateien sind jeweils der technischen Dokumentation der Basisspezifikation für Leistungserbringer bzw. Datenannahmestellen zu entnehmen.

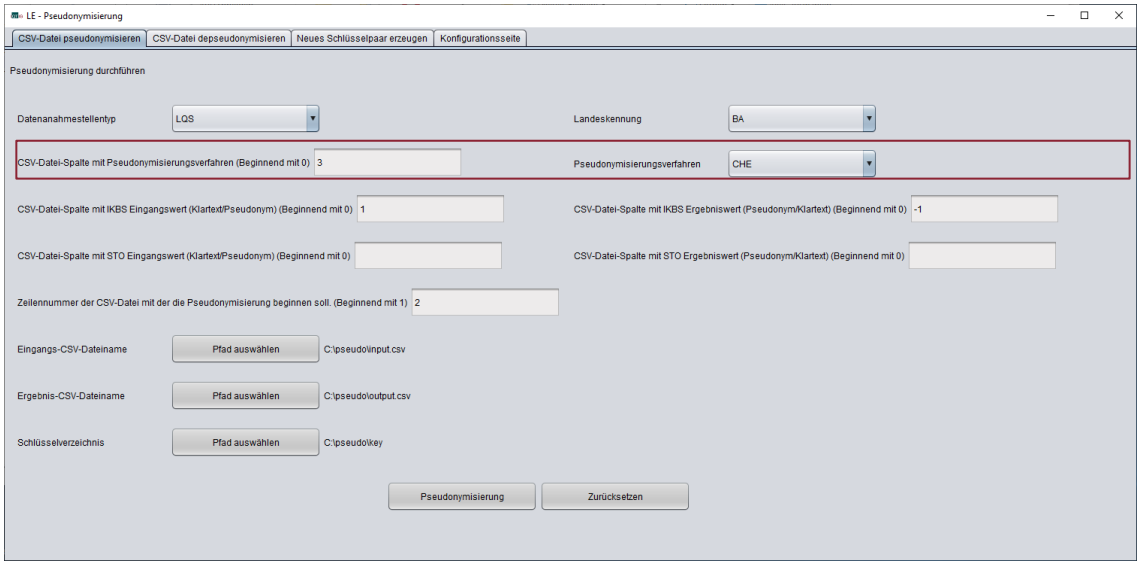

Abbildung 3: Oberfläche zur Pseudonymisierung der LE-Informationen

Beim Klick auf die Schaltfläche „Zurücksetzen“ werden die Formularwerte auf die Werte der Konfigurationsseite zurückgesetzt.

3.1.4 Durchführung der Depseudonymisierung

Analog zur Pseudonymisierung nutzt das Pseudonymisierungsprogramm denselben Algorithmus zur Depseudonymisierung der Leistungserbringerpseudonyme. Die Depseudonymisierung mit dem Pseudonymisierungsprogramm ist nur durch die LQS/LKG/KV/KZV durchzuführen. Die DAS-KK und DAS-SV können hingegen nur auf Mapping-Informationen zurückgreifen. Die Depseudonymisierung der LE-Pseudonyme in den Rückprotokollen bzw. in den Rückmeldeberichten ist gemäß Richtlinie erforderlich.

Die Oberfläche zur Depseudonymisierung ist in Abbildung 4 dargestellt. Eine detaillierte Beschreibung der entsprechenden Auswahlfelder befindet sich in Tabelle 1.

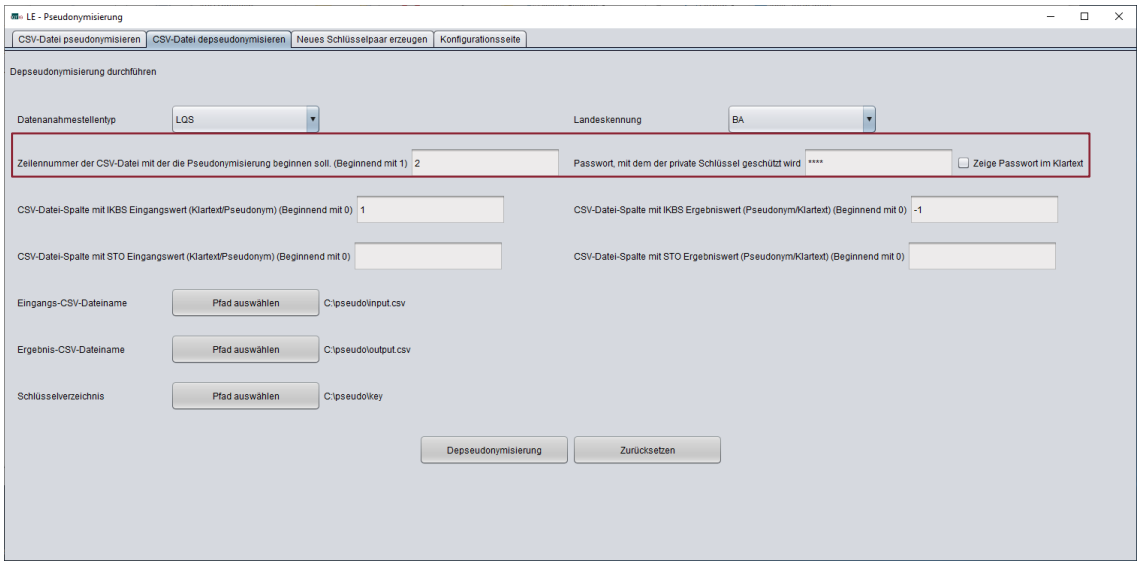

Abbildung 4: Oberfläche zur Depseudonymisierung der LE-Informationen

Beim Klick auf die Schaltfläche „Zurücksetzen“ werden die Formularwerte auf die Werte der Konfigurationsseite zurückgesetzt.

3.2 Konsolenanwendung

Die Datenannahmestellen können das Pseudonymisierungsprogramm als Konsolenapplikation in eine „Nicht-Java“-Datenverarbeitungskette integrieren. Das Programm wird in diesem Fall dafür benutzt, eine CSV-Datei mit einer Liste von LE-Daten zu pseudonymisieren bzw. zu depseudonymisieren. Hierbei wird ein System-Return-Code erzeugt, der die jeweiligen Erfolgs- oder Fehlerfälle ausgibt. Durch Konsolen-Parameter oder wahlweise eine Eigenschaftsdatei (Properties-File) erhält das Pseudonymisierungsprogramm sowohl die Information über den Zugriffsort der CSV-Datei als auch über den Speicherort der Ergebnisdatei.

Abhängig von den durchzuführenden Aktionen ist die Kommandozeile entsprechend anzupassen. Tabelle 3 gibt einen Überblick über die einzelnen Anwendungsfälle.

Tabelle 3: Anwendungsfälle der Konsolenanwendung

Use-Case	Kommandozeile
Generierung der Zertifikate	`java -jar pseudonymisierung-console.jar -a LQS -b 30.06.2015 -c C:\pseudo\key -l BW -s test -z`
Pseudonymisierung	`LE` `java -jar pseudonymisierung-console.jar -a LQS -c C:\pseudo\key -l NI -i C:\pseudo\input.csv -j 2 -k 1 -p -r C:\pseudo\out\outk.csv -t "-1" -w PCI STO`

Use-Case	Kommandozeile
	`java -jar pseudonymisierung-console.jar -a LQS -c C:\pseudo\key -l NI -i C:\pseudo\input.csv -j 2 -k 1 -p -r C:\pseudo\out\outk.csv -t "-1" -w PCI`
Depseudonymisierung	`java -jar pseudonymisierung-console.jar -a LQS -c C:\pseudo\key -d -l NI -i C:\pseudo\depseu.csv -j 2 -k 1 -r C:\pseudo\out\outdepseuk.csv -t "-1" -s test`
Übermittlung der Zusatzdaten an die BAS	`java -jar pseudonymisierung-console.jar -h -a LQS -c C:\pseudo\key -l NI -I C:\pseudo\pseudonymMappings.csv -m C:\pseudo\zusatzDaten.csv -n username -q password`

Tabelle 4: Parameter der Konsolenanwendung

Parameter	Bedeutung
`-a,--das <arg>`	Datenannahmestelle, deren Zertifikate benutzt werden sollen. Gültige Werte sind LQS, BAS, KV, SV (siehe Anhang).
`-b,--sdate <arg>`	Beginn der Gültigkeit des zu erzeugenden Schlüssel-Zertifikats.
`-c,--certificates <arg>`	Verzeichnis für die Schlüsseldateien.
`-d,--depseudo`	Startet das Programm im Depseudonymisierungs-Modus.
`-e,--edate <arg>`	Ende der Gültigkeit des zu erzeugenden Schlüssel-Zertifikats.
`-f,--cert <arg>`	Zertifikat, welches verwendet werden soll.
`-i,--input <arg>`	CSV-Datei, welche pseudonymisiert bzw. depseudonymisiert werden soll. CSV-Datei, welche pro Zeile eine Liste von zusammengehörenden IKNR/BSNR enthält, für den Fall, dass Zusatzdaten an die BAS übertragen werden sollen.
`-j,--rowStart <arg>`	Zeile der CSV-Datei, ab der die Datei bearbeitet werden soll (Beginnend mit 1).
`-k,--inputSpalte <arg>` (deprecated)	Spalte in der CSV-Datei mit den Eingangsdaten der Leistungserbringer (Klartext bei Pseudonymisierung, Pseudonym bei Depseudonymisierung), beginnend mit 0.
`-kib,--input-SpalteIkbs <arg>`	Spalte in der CSV-Datei mit den Eingangsdaten der IKNR/BSNR (Klartext bei Pseudonymisierung, Pseudonym bei Depseudonymisierung), beginnend mit 0. (Entspricht dem Parameter –k / --inputSpalte)
`-ks,--inputSpalteSto <arg>`	Spalte in der CSV-Datei mit den Eingangsdaten der Standorte-ID (Klartext bei Pseudonymisierung, Pseudonym bei Depseudonymisierung), beginnend mit 0.

Parameter	Bedeutung
`-l,--Länderkürzel <arg>`	Länderkürzel (DAS-LE-Identität); siehe Tabelle 6
`-o,--properties <arg>`	Dateipfad der zu benutzenden Properties-Datei.
`-p,--pseudo`	Startet das Programm im Pseudonymisierungsmodus.
`-r,--output <arg>`	Ziel der Ergebnis-CSV-Datei
`-s,--certpassword <arg>`	Passwort der Zertifikatsdatei
`-t,--outputSpalte <arg>` (deprecated)	Die Spalte der CSV-Datei, in welche der Ergebniswert des Pseudonymisierungsvorganges der IKNR/BSNR eingetragen wird. Ist der Wert größer als die Anzahl der Spalten in der CSV-Datei, so wird das Ergebnis als letzte Spalte in die CSV-Datei eingetragen. Ist der Wert kleiner als 0, so wird der Eingangswert mit dem Ergebnis überschrieben.
`-tib,--output-SpalteIkbs <arg>`	Die Spalte der CSV-Datei, in welche der Ergebniswert des Pseudonymisierungsvorganges für die IKNR/BSNR eingetragen wird. Ist der Wert größer als die Anzahl der Spalten in der CSV-Datei, so wird das Ergebnis als letzte Spalte in die CSV-Datei eingetragen. Ist der Wert kleiner als 0, so wird der Eingangswert mit dem Ergebnis überschrieben. (Entspricht dem Parameter –t / --outputSpalte)
`-ts,--outputSpalteSto <arg>`	Die Spalte der CSV-Datei, in welche der Ergebniswert des Pseudonymisierungsvorganges der Standort-ID eingetragen wird. Ist der Wert größer als die Anzahl der Spalten in der CSV-Datei, so wird das Ergebnis als letzte Spalte in die CSV-Datei eingetragen. Ist der Wert kleiner als 0, so wird der Eingangswert mit dem Ergebnis überschrieben.
`-v,--verfahrenSpalte <arg>`	Spalte in der CSV-Datei mit Verfahrenskürzel
`-w,--verfahren <arg>`	Verfahrenskürzel, welches benutzt werden soll.
`-z,--creaCert`	Erzeugt ein neues Zertifikat.
`-h, --send`	Sendet die aktuellen Einträge der Zusatzdatenverwaltung an die BAS.
`-m, --addData <arg>`	Die Zusatzdaten-CSV-Datei.

Parameter	Bedeutung
`-n, --user <arg>`	Der Benutzername bei der BAS.
`-q, --password <arg>`	Das Benutzerpasswort bei der BAS.

4 Fehlermeldungen

Im Folgenden werden alle Fehlermeldungen beschrieben, die während der Nutzung des Programms ausgegeben werden können.

Sollte das Programm bei einem Doppelklick nicht starten oder das „Öffnen mit“-Fenster von Windows geöffnet werden, ist möglicherweise die Java-Laufzeitumgebung nicht korrekt installiert. In diesem Fall wenden Sie sich bitte an Ihren Systemadministrator.

Geschweifte Klammern { } in der Spalte „Bedeutung“ von Tabelle 5 werden vom Programm dynamisch sinnvoll ersetzt.

Beispiel:
`Bei der Verarbeitung der CSV-Datei {} ist ein Fehler in Zeile {} aufgetreten.`

Wird ersetzt durch:

`Bei der Verarbeitung der CSV-Datei c:/temp/Muster.csv ist ein Fehler in Zeile 25 aufgetreten.`

Tabelle 5: Fehlercodes

Fehlercode	Fehlermeldung	Erläuterung
1000	Unbekannt	
1	Konnte `certificate.startdate` nicht konvertieren.	`certificate.startdate` ist im falschen Format angegeben.
2	Konnte keine Schlüsselpaare erzeugen.	
4	Konnte den öffentlichen Schlüssel nicht schreiben.	Auf das angegebene ‚Schlüssel Verzeichnis‘ (`registration.dir`) konnte nicht schreibend zugegriffen werden.
5	Konnte den PKCS12-Schlüssel nicht schreiben.	Auf das angegebene ‚Schlüssel Verzeichnis‘ `(registration.dir)` konnte nicht schreibend zugegriffen werden.
6	Konnte den öffentlichen Schlüssel nicht lesen.	
7	Der Schlüssel existiert schon.	-
9	Unbekannter Zertifikatstyp	-

Fehler-code	Fehlermeldung	Erläuterung
10	Die im Schlüssel enthaltenen Metadaten enthalten falsche Angaben zur Datenannahmestelle.	-
12	Die Daten konnten nicht pseudonymisiert werden.	
13	Die Daten konnten nicht depseudonymisiert werden.	
14	Falscher Dateiname für den öffentlichen Schlüssel.	-
15	Registrationsverzeichnis existiert nicht.	-
16	Das angegebene Registrationsverzeichnis ist kein Verzeichnis.	-
17	Fehler beim Erzeugen des privaten Schlüssels.	
18	Fehler beim Laden des privaten Schlüssels. Entweder das Passwort ist falsch oder das Zertifikat ist defekt.	-
19	Der Klartext {} konnte nicht vom Verfahren getrennt werden.	Das gerade depseudonymisierte Pseudonym ist aus einem nicht der Spezifikation entsprechendem Klartext erzeugt worden. Bitte benachrichtigen Sie die BAS über diesen Vorfall.
20	Die Property mit dem Namen {} wird benötigt.	-
21	Die Property mit dem Namen {} muss einen Integer-Wert enthalten.	-
22	CSV-Datei {} konnte nicht gefunden werden.	-
26	Es gibt keine Länderkürzel mit dem Namen oder Kürzel {}.	-
27	Es gibt kein Modul mit dem Kürzel {}.	-
28	Es gibt kein Verfahren mit dem Kürzel {}.	-
29	Es gibt keine Datenannahmestelle mit dem Anfangskürzel {}.	-
30	Keine Schlüssel für die Datenannahme {} {} im Verzeichnis {} gefunden.	-

Fehler-code	Fehlermeldung	Erläuterung
32	In der LQS kann kein Bundesland {} ausgewählt werden.	-
33	In der KV kann kein Bundesland {} ausgewählt werden.	-
34	Die CSV-Datei hat nur {} Spalten in {} wurde {} angegeben.	-
35	Es wurde kein Zertifikat gefunden, welches zum angegebenen Datum {} valid ist.	-
36	Die benutzte Datumsformatierung ist nicht valid {} muss dd.MM.yyyy genügen.	-
37	Das Pseudonym {} konnte nicht von der Zertifikats-ID getrennt werden.	Das angegebene Pseudonym entspricht nicht der Spezifikation. Bitte benachrichtigen Sie die BAS über diesen Vorfall.
38	Die Zertifikats-ID {} ist invalid.	Die angegebene Zertifikats-ID entspricht nicht der Spezifikation. Bitte benachrichtigen Sie die BAS über diesen Vorfall.
39	Das Pseudonym wurde von einer anderen Datenannahmestelle pseudonymisiert. Die Zertifikats-ID der fremden DAS lautet {}.	-
40	Zeile {} in CSV-Datei {} ist nicht korrekt formatiert.	-
41	In der CSV-Datei {} in der Zeile {} fehlt ein Hauptpseudonym.	-
42	Bei der Übertragung der Daten an die BAS ist ein Fehler aufgetreten: {}	
43	Bei der Verarbeitung der CSV-Datei {} ist ein Fehler in Zeile {} aufgetreten.	
44	Allgemeiner Fehler beim Laden des Keystores.	
1001	Konsolen-Argument 'p', 'd' oder 'z' erforderlich!	-
1002	Konsolen-Argument {} erforderlich!	-

Bei der Verwendung der grafischen Benutzeroberfläche oder des Konsolenprogramms wird im gleichen Verzeichnis eine `pseudonymisierungs.log`-Datei geschrieben. In dieser Datei werden Programmvorgänge und Fehler protokolliert. Die dieses Protokoll beinhaltende Datei ist der Schlüssel zur Fehleranalyse.

Wird die Referenzimplementierung verwendet, sollte das Slf4j-Logging-Framework konfiguriert werden, damit auch hier die entsprechenden Informationen protokolliert werden.

Die meisten möglichen Fehlermeldungen sind selbsterklärend. Wird auf dieses Kapitel verwiesen, macht ein Blick in diese Logdatei Sinn.

Sofern die Fehleranalyse von der Datenannahmestelle nicht allein bewältigt werden kann, sollte diese Datei nach Auftreten des Fehlers gesichert und bei Unterstützungsanfragen an die BAS mitgesendet werden.

5 Änderungen der Version 05 / PSP 2.0.15 (Details)

5.1 Funktionalität

Es ist jetzt möglich Leistungserbringerpseudonyme zu erzeugen, die entweder auf der IKNR bzw. Betriebsstättennummer (BSNR) oder auf der Standort-Id (STOID) beruhen. Optisch können diese Pseudonyme anhand ihres Präfixes unterschieden werden.

Für Pseudonyme auf Basis der IKNR / BSNR wird weiterhin die bisher übliche *CertificateIdentity* Id verwendet, die sich aus einem Kürzel der Daten entgegennehmenden Stelle (z.B. LQS, KV) und dem StateCode (kennzeichnet eine Region) zusammensetzt. Hierbei wird das Kürzel der Daten entgegennehmenden Stelle und der StateCode durch zwei Dollar-Zeichen ($$) voneinander separiert.

Bsp: `LQS$BA…`

Für Pseudonyme auf Basis der Standort-Id wird das Dollar-Trennzeichen durch zwei Sterne (**) ersetzt, so dass erkennbar wird um was für ein Pseudonym es sich handelt.

Bsp: `LQ**BA…`

Beide Präfixe werden mit dem Base64 encodierten Chiffretext durch einen Doppelten Hashtag (##) verbunden.

Bsp:

IKNR: `LQS$$BA##XOseAM4Vp318ZPKi/bOGch23OKcj1lR5sqOR...`

STO: `LQS**BA##4ZKcYiada2Ts+AqFKfiXIsamjIB1em+MR763tN...`

5.2 Entfernte Maven-Module

Folgende Maven Module wurden aus dem Projekt entfernt.

```
pseudonymisierung-zus-cli
pseudonymisierung-zus-model
pseudonymsierung-zus-ser
```

Bei diesen handelte es sich um Funktionalität um Zusatzdaten an die BAS (IQTIG) zu übertragen. Da dies auf anderem Weg, als über einen SOAP-Webservice erfolgen wird konnten die überflüssigen Module entfernt werden.

5.3 Console - Modul

5.3.1 Neue CLI Paramter

```
--inputSpalteIkbs, -kib
--outputSpalteIkbs, -tib
```

```
--inputSpalteSto, -ks
--outputSpalteSto, -ts
```

5.3.2 Entfernte CLI Parameter

Aufgrund der Entfernung der Module für die Zusatzdaten wurden die folgenden Parameter überflüssig und konnten entfernt werden.

```
--send, -h, Sende aktualisierte Einträge an die BAS
--user, Username bei der BAS
--password, Passwort bei der BAS
```

5.3.3 Deprecated Parameter

Die bisherigen Parameter zur Kennzeichnung der Input und Output Spalte in der CSV Datei im Zusammenhang mit der Pseudonymisierung wurden *Deprecated* markiert und können zukünftig entfernt werden.

Deprecated	Ersetzt durch
`--inputSpalte`	`--inputSpalteIkbs, -kib`
`--outputSpalte`	`--outputSpalteIkbs, tib`

5.4 API / Impl - Modul

Die bisherigen Klassen (CsvPseudonymisator, CsvDepseudonymisator) zur Verarbeitung von CSV Dateien zur Pseudonymisierung und Depseudonymisierung wurden angepast, so dass sie mit den neuen Properties (COLUMN_INPUT_IKBS, COLUMN_OUTPUT_IKBS, COLUMN_INPUT_STO, COLUMN_OUTPUT_STO) zur Pseudonymbehandlung umgehen können.

Um auf einem direkteren Weg Pseudonyme erzeugen und entschlüsseln zu können sind zwei neue Klassen geschaffen worden, die hinsichtlich ihrer Api unabhängig von Dateipfaden oder Keystore-Dateien verwendet werden können.

```
IkbsAndStoIdPseudonymizator
   Methode: String createIkbsPseudonym(String ikbs, Method pseudony-
   misierungsVerfahren)
   Methode: String createStoPseudonym(String stoId, Method pseudony-
   misierungsVerfahren)
IkbspAndStopDepseudonymizator
   Methode: PseudonymTO depseudonymize(String pseudonym)
```

5.5 Neuer Fehlercode

Im Zusammenhang mit Änderungen in der *org.iqtig.pseudonymisierung.impl.helper.FileHelper#readPkcs12KeyStore* Methode entstand ein neuer Fehlercode.

```
READ_PKCS12_COMMON(44, "Allgemeiner Fehler beim Laden des Key-
stores.")
```

Dieser kennzeichnet einen allgemeinen Fehler, wenn ein PKCS12 Keystore geladen werden soll und dies fehlschlägt. Daraus u.U. resultiert im Fehlerfall ein verändertes Verhalten der Anwendung.

Bisher:

Bisher wurden Fehler die beim Laden des Keystores auftreten können in den meisten Fällen ignoriert und mit der Anwendung fortgefahren. Das resultierte in einem potentiellen Verschlucken von Fehlern. Nur in bestimmten Fällen wurden die Fehler weitergereicht und dem Benutzer signalisiert, z.B. falsches oder kein Passwort gesetzt. Dabei wurden Annahmen über das JRE mit welchem die Anwendung gestartet wurde getroffen. Da die Verwendung des JREs in der Außenwelt nicht kontrollierbar ist machte der Umbau Sinn.

Zu den verschluckten Fehlerursachen gehörten zum Beispiel defekte PKCS12 Keystores im Zertifikats-Dateiordner.

Neu:

Es wir kein Fehler, der bei Öffnen / Laden eines Keystores auftritt mehr verschluckt. Das bedeutet, dass defekte Keystores oder Keystores mit beschädigten Inhalten (Keys / Zertifkate) zu einem Abbruch der Anwendung führen, so dass der Benutzer diese Fehler korrigieren kann. Ein Manko besteht in der Interpretation von falsches / kein Passwort Fehlern. Theoretisch kann es sein, dass ein anderer Fehler als ErrorCode *READ_PKCS12* ausgegeben wird. Das liegt daran, dass auf die *UnrecoverableKeyException* des JREs mehrere Fehler vom JRE selbst abgebildet werden, wodurch nicht ganz eindeutig der Fall falsches / kein Passwort bestimmt werden kann. Da das Javadoc der JRE *Keystore.load* Methode aber folgendes besagt,

```
@exception IOException if there is an I/O or format problem with
the
     * keystore data, if a password is required but not given,
     * or if the given password was incorrect. If the error is due
to a
     * wrong password, the {@link Throwable#getCause cause} of the
     * {@code IOException} should be an
     * {@code UnrecoverableKeyException}
```

ist anzuenhmen, dass ein falsches oder kein Passwort der gängige Fall ist und man die *UnrecoverableKeyException* entsprechend wie umgesetzt interpretieren kann.

Fehlercode: READ_PKCS12(18, "Fehler beim Laden des privaten Schlüssels. Entweder das Passwort ist falsch oder das Zertifikat ist defekt.")

5.6 GUI - Modul

Aus dem GUI Modul wurden alle entfernten Config-Parameter entfernt.

Die deprecated Parameter *COLUMN_INPUT* und *COLUMN_OUTPUT* wurden in der Oberfläche durch die Parameter *COLUMN_INPUT_IKBS* und *COLUMN_OUTPUT_IKBS* ersetzt, da diese die

gleiche Funktionalität gewährleisten. Das bedeutet, dass Benutzer ihre *pseudonym-config.properties* Datei, die sich in ihrem User-Home Verzeichnis befindet neu erzeugen oder anpassen müssen.

5.7 Beispiele - CLI

Die CLI-Beispiele entsprechen den Möglichkeiten, die man über die GUI hat.

5.7.1 Erzeugung eines Pseudonyms auf Basis der IKNR / BSNR (bisher)

```
java -jar pseudonymisierung-console-2.0.12-SNAPSHOT.jar -a LQS -c
key -l BA -i input.csv -j 2 -k 1 -p -r out.csv -t "-1" -w PCI
```

CSV-Input:

```
irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgend-
welcheDaten
LoremYpsum;999999999;LoremYpsum;PCI;LoremYpsum
```

CSV-Output:

```
irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgend-
welcheDaten
LoremYpsum;LQS$$BA##XOseAM4Vp318ZPKi/bOGch23OKcj1lR5sqORR-
kny1c+TYs8ZTwuncTg7hYgGpWmo9yJh63WUoO1Y293okAkrkPuM6UbKC3JVDkhk8g
P0qTyW1ACI7V/vvtjseJoMc3vCMZW4ZKcYiada2Ts+AqFKfiXI-
samjIB1em+MR763tNIqKnyz88FscYD6Y3uHhY+EwJDHIu1Osn-
Mecg+k5ckX2U1prYs+pqPpZhsRJbZel43KovLGezDC0/5AMFCyIOT-
vguVCVlVT9jHB2aFmhRpbpfYQhhv5mj5G+Y4Ebfp67Q+8uwcOO2qw4s5CXa2EnNTA
CJYLzM76nimMRme3mnoi7nA==;LoremYpsum;PCI;LoremYpsum
```

5.7.2 Erzeugung eines Pseudonyms auf Basis der IKNR / BSNR (neu)

```
java -jar pseudonymisierung-console-2.0.12-SNAPSHOT.jar -a LQS
-c key -l BA -i input.csv -j 2 -kib 1 -p -r out.csv -tib "-1" -
w PCI
```

CSV-Input:

```
irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgend-
welcheDaten
LoremYpsum;999999999;LoremYpsum;PCI;LoremYpsum
```

CSV-Output:

```
irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgend-
welcheDaten
LoremYpsum;LQS$$BA##XOseAM4Vp318ZPKi/bOGch23OKcj1lR5sqORR-
kny1c+TYs8ZTwuncTg7hYgGpWmo9yJh63WUoO1Y293okAkrkPuM6UbKC3JVDkhk8g
P0qTyW1ACI7V/vvtjseJoMc3vCMZW4ZKcYiada2Ts+AqFKfiXI-
samjIB1em+MR763tNIqKnyz88FscYD6Y3uHhY+EwJDHIu1Osn-
Mecg+k5ckX2U1prYs+pqPpZhsRJbZel43KovLGezDC0/5AMFCyIOT-
vguVCVlVT9jHB2aFmhRpbpfYQhhv5mj5G+Y4Ebfp67Q+8uwcOO2qw4s5CXa2EnNTA
CJYLzM76nimMRme3mnoi7nA==;LoremYpsum;PCI;LoremYpsum
```

5.7.3 Erzeugung eines Pseudonyms auf Basis der Standort-Id (StoId)

```
java -jar pseudonymisierung-console-2.0.12-SNAPSHOT.jar -a KV -c
key -l BA -i input.csv -j 2 -ks 1 -p -r out.csv -ts "-1" -w PCI
```

CSV-Input:

irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgendwelcheDaten

LoremYpsum;771234;LoremYpsum;PCI;LoremYpsum

CSV-Output:

```
irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgend-
welcheDaten
LoremYpsum;LQS**BA##XOseAM4Vp318ZPKi/bOGch23OKcj1lR5sqORR-
kny1c+TYs8ZTwuncTg7hYgGpWmo9yJh63WUoO1Y293okAkrkPuM6UbKC3JVDkhk8g
P0qTyW1ACI7V/vvtjseJoMc3vCMZW4ZKcYiada2Ts+AqFKfiXI-
samjIB1em+MR763tNIqKnyz88FscYD6Y3uHhY+EwJDHIu1Osn-
Mecg+k5ckX2U1prYs+pqPpZhsRJbZel43KovLGezDC0/5AMFCyIOT-
vguVCVlVT9jHB2aFmhRpbpfYQhhv5mj5G+Y4Ebfp67Q+8uwcOO2qw4s5CXa2EnNTA
CJYLzM76nimMRme3mnoi7nA==;LoremYpsum;PCI;LoremYpsum
```

5.7.4 Erzeugung eines Pseudonyms auf Basis der IKNR / BSNR, sowie der Standort-Id (StoId)

```
java -jar pseudonymisierung-console-2.0.13-SNAPSHOT.jar -a KV -c
key -l BA -i input-multiple.csv -j 2 -kib 1 -ks 5 -p -r out-mul-
tiple.csv -tib "-1" -ts "-1" -w PCI
```

CSV-Input:

```
irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgend-
welcheDaten;stoid;irgendwelcheDaten
LoremYpsum;123456789;LoremYpsum;PCI;LoremYpsum;771234;LoremYpsum
```

CSV-Output:

```
irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgend-
welcheDaten;stoid;irgendwelcheDaten
LoremYpsum;KV$$BA##XOseAM4Vp318ZPKi/bOGch23OKcj1lR5sqORR-
kny1c+TYs8ZTwuncTg7hYgGpWmo9yJh63WUoO1Y293okAkrkPuM6UbKC3JVDkhk8g
P0qTyW1ACI7V/vvtjseJoMc3vCMZW4ZKcYiada2Ts+AqFKfiXI-
samjIB1em+MR763tNIqKnyz88FscYD6Y3uHhY+EwJDHIu1Osn-
Mecg+k5ckX2U1prYs+pqPpZhsRJbZel43KovLGezDC0/5AMFCyIOT-
vguVCVlVT9jHB2aFmhRpbpfYQhhv5mj5G+Y4Ebfp67Q+8uwcOO2qw4s5CXa2EnNTA
CJYLzM76nimMRme3mnoi7nA==;LoremYpsum;PCI;LoremYp-
sum;KV**BA##atNgZ1r65QWFhxLIRbtRpfzHItKh8OJvs85w6FXxdQc13g5w+OK-
SenAT++3LM61DtBAEFcp7e-
cRQrHW/b6bs3d1/RnrqCU8hM12zlIyhvbSBW9r0DumyfiQh9zER4Ipt+t4NAOVtnM
f3FNWc72lj98QRVm7P5Mmx7O200lfNfeqeT1c/C7iPMp4lTJFS-
ssPxBx2v32RJ3yT0shXdTNwiESeXS85rpeJahT1PayGuen-
rMs6oqjrAXs+MYjbLBa3nUqStarRzjO8x1Y56YeS9BMuQ6Yv5SVDiYSe/G21H1iKN
iWFOrw5aABWbIvg5ozVaF2+E9/okcRXaErs8cRJYUVw==;LoremYpsum
```

5.7.5 Depseudonymisierung von Pseudonymen auf Basis der IKNR / BSNR und der Standort-id (StoId)

```
java -jar pseudonymisierung-console-2.0.13-SNAPSHOT.jar -a LQS -c
key -d -l BA -i depseudonymization-ikbsp-stop.csv -j 2 -kib "-1" -
ks "-1" -r out-depseudo.csv -tib "-1" -ts "-1" -v 3 -s test
```

CSV-Input:

```
irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgend-
welcheDaten;stoid;irgendwelcheDaten
LoremYpsum;LQS$$BA##XOseAM4Vp318ZPKi/bOGch23OKcj1lR5sqORR-
kny1c+TYs8ZTwuncTg7hYgGpWmo9yJh63WUoO1Y293okAkrkPuM6UbKC3JVDkhk8g
P0qTyW1ACI7V/vvtjseJoMc3vCMZW4ZKcYiada2Ts+AqFKfiXI-
samjIB1em+MR763tNIqKnyz88FscYD6Y3uHhY+EwJDHIu1Osn-
Mecg+k5ckX2U1prYs+pqPpZhsRJbZel43KovLGezDC0/5AMFCyIOT-
vguVCVlVT9jHB2aFmhRpbpfYQhhv5mj5G+Y4Ebfp67Q+8uwcOO2qw4s5CXa2EnNTA
CJYLzM76nimMRme3mnoi7nA==;LoremYpsum;PCI;LoremYp-
sum;KV**BA##atNgZ1r65QWFhxLIRbtRpfzHItKh8OJvs85w6FXxdQc13g5w+OK-
SenAT++3LM61DtBAEFcp7e-
cRQrHW/b6bs3d1/RnrqCU8hM12zlIyhvbSBW9r0DumyfiQh9zER4Ipt+t4NAOVtnM
f3FNWc72lj98QRVm7P5Mmx7O200lfNfeqeT1c/C7iPMp4lTJFS-
ssPxBx2v32RJ3yT0shXdTNwiESeXS85rpeJahT1PayGuen-
rMs6oqjrAXs+MYjbLBa3nUqStarRzjO8x1Y56YeS9BMuQ6Yv5SVDiYSe/G21H1iKN
iWFOrw5aABWbIvg5ozVaF2+E9/okcRXaErs8cRJYUVw==;LoremYpsum
```

CSV-Output:

```
irgendwelcheDaten;iknr;irgendwelcheDaten;pseudoVerfahren;irgend-
welcheDaten;stoid;irgendwelcheDaten
LoremYpsum;123456789;LoremYpsum;PCI;LoremYpsum;771234;LoremYpsum
```

Anhang

Tabelle 6: Gültige Länderkürzel

Kürzel	Bedeutung
BU	Bundesweit
BA	Bayern
BB	Brandenburg
BE	Berlin
BW	Baden-Württemberg
HB	Bremen
HE	Hessen
HH	Hamburg
MV	Mecklenburg-Vorpommern
NI	Niedersachsen
NO	Nordrhein
NW	Nordrhein-Westfalen
RP	Rheinland-Pfalz
SH	Schleswig-Holstein
SL	Saarland
SN	Sachsen
ST	Sachsen-Anhalt
TH	Thüringen
WL	Westfalen-Lippe
TESTLAND	Zur Nutzung auf den Testdatenstrecken zwischen Softwareanbietern und Datenannahmestellen

Tabelle 7: Gültige Datenannahmestellen

Kürzel	Datenannahmestelle
LQS	Landesgeschäftsstelle für Qualitätssicherung
BAS	Bundesauswertungsstelle
KV	Kassenärztliche Vereinigung
SV	Datenannahmestelle für selektivvertraglich erbrachte Leistungen