EUROPÄISCHE UNION
Investition in unsere Zukunft
Europäischer Fonds
für regionale Entwicklung

Die Landesregierung
Nordrhein-Westfalen

eEFA Düren

Bereitstellung und zentrale Auswertung von medizinischen und ökonomischen Daten der Netzärzte

Fachkonzept Controlling Anhang Pseudonymisierung

Kassenärztliche Vereinigung Nordrhein

Deutsches Gesundheitsnetz Service GmbH

brightONE GmbH

Krankenhaus Düren eGmbH

Duria eG

DAGIV eG

Version: 1.00

Stand: 25.04.2014

Inhaltsverzeichnis

Einleitung

Das Controlling-Projekt soll es den am Projekt beteiligten Ärzten ermöglichen, aus den von ihnen erzeugten validen Behandlungsdaten Informationen zur Verbesserung der Behandlungsprozesse abzuleiten. Die Verbesserung der Behandlungsprozesse betrifft in diesem Kontext sowohl die Optimierung der medizinischen Prozeduren als auch in der direkten Folge die Reduktion des erforderlichen Aufwandes. Der erste Schritt zu diesem Vorgehen ist das Sammeln von Behandlungsdaten und das sofortige datenschutzrechtlich einwandfreie Aufarbeiten in Form der Überführung der Daten in pseudonyme/anonyme Form vor der weiteren Verarbeitung. Ein Teil des Fachkonzepts Controlling beschreibt auf niedrigem technischen Niveau die Verarbeitung der Daten, die für pseudonymisierte Auswertungen genutzt werden sollen. Zur Beurteilung des Verfahrens aus datenschutzrechtlicher Sicht und zur Realisierung der Nachvollziehbarkeit ist aber eine höhere Detaillierung erforderlich. Das vorliegende Dokument dient als Anlage zum Fachkonzept Controlling diesem Zweck.

1 Voraussetzungen

Um sinnvoll mit nicht mehr personenbezogenen Daten Auswertungen durchführen zu können, müssen die Rohdaten zuerst erfasst und gespeichert werden. Von den Stellen, an denen dies rechtskonform durchgeführt wird, muss sichergestellt werden, dass Daten, die in die Auswertung einfließen, den konkreten Personen nicht mehr zugeordnet werden können. Um aus pseudonymisierten Daten sinnvolle Ergebnisse ableiten zu können, sind zwei Voraussetzungen erforderlich:

1 eine sinnvolle Fragestellung
2 eine große Datenmenge

Um medizinische Fragestellungen zu untersuchen, ist die Sammlung medizinischer Behandlungsdaten erforderlich. Diese Daten werden rechtskonform vor allem (nur?) bei Gesundheitsdienstleistern (Health Professionals) und hier in der Regel von Ärzten bzw. ärztlichen Einrichtungen erfasst und verarbeitet. Diese bieten sich als Datenquellen an.

Die bei den Ärzten erfassten Daten müssen vor der Zusammenführung und Weiterverarbeitung zur Wahrung der Patientenrechte anonymisiert/pseudonymisiert werden. Je nach Ziel der Auswertungen kann vor allem die Pseudonymisierung datenschutz-

rechtlich recht unterschiedlich gestaltet werden. Wichtigstes Kriterium dabei ist, ob später aus verschiedenen Gründen eine potentielle Repersonalisierung der Daten vorgesehen werden soll.

Im vorliegenden Fall ist das nicht vorgesehen, so dass man den datenschutzfreundlichsten Fall der Pseudonymisierung realisieren kann, die Pseudonymisierung ohne Rückführbarkeit. Im konkreten Fall erfolgt diese Pseudonymisierung nicht nur patientenbezogen, auch die liefernden Einrichtungen werden „unkenntlich“ gemacht.

Dieses Vorgehen muss, wie bei allen diesen Verfahren, flankiert werden durch eine Filterung der Eingangsdaten, um immanent zur Identifizierung geeignete Daten nicht in den Prozess gelangen zu lassen. Weiterhin benötigt dieses Verfahren ein sorgfältiges Design der Auswertungen, so dass eine hinreichende k-Anonymität für jedes Auswerteszenario gesichert wird. Neben den technischen Voraussetzungen ist deshalb die konzeptionelle, nichttechnische Organisation der Auswertung im Vorfeld essentiell. Dieser nichttechnische Teil der pseudonymen Datenauswertung unterbleibt in diesem Dokument.

Die Pseudonymisierung ohne die Erfordernis der Repersonalisierung lässt eine im Vergleich zu anderen Methoden sehr vereinfachte Infrastruktur bei verbessertem Datenschutz zu. Ihre Realisierung erfolgt im vorliegenden Fall durch eine (mutmaßlich) neue Methode, die mit bewährten kryptographischen Algorithmen implementiert ist.

2 Realisierung

Die Realisierung erfolgt, sehr abstrakt betrachtet, in einer „Three Tier Architecture“. Die drei beteiligten „Tier“ sind die medizinische Einrichtung, die Pseudonymisierungsstelle (TTP) und die Auswertestelle. Die drei „Tier“ sind sehr lose gekoppelt. Die Kopplung erfolgt über die Kommunikation über D2D.

Die Infrastruktur ist in der Lage, eine weitgehend unbegrenzte Anzahl verschiedener Pseudonymisierungsmodelle zu betreiben. Dabei könnte eines der Modelle Versorgungsforschung betreiben, ein anderes medizinische Forschung zu einem speziellen Krankheitsbild, ein weiteres den Ressourcenverbrauch einer medizinischen Fachrichtung usw..

Basis eines jeden Modells ist ein eigener Input-Filter (eine Datei, die beim Einlieferer, also i.A. dem Arzt, eine Liste der nicht benötigten Items beschreibt), einen D2D-ID für eine TTP (Trusted Third Party oder Pseudonymisierungsstelle) und eine D2D-ID für die Auswertestelle. Weiterhin sind für die TTP und die Auswertestelle jeweils ein Satz DH-Schlüssel (Schlüsselpaar für das Diffie-Hellmann-Verfahren) zu erstellen und der öffentliche DH-Schlüssel der Auswertestelle ist an die Datenlieferer zu verteilen.

2.1 Datengewinnung

Die Datengewinnung setzt im vorliegenden Fall dort an, wo die Daten entstehen, also bei den niedergelassenen Ärzten, die an der Maßnahme teilnehmen. Ärzte nutzen im Normalfall ein sogenanntes PVS (Praxisverwaltungssystem) oder AIS (Arzt-Informationssystem), eine Software, deren Funktionsumfang je nach Wahl des Produkts oder finanziellem Engagement von der reinen Erfassung der abrechenbaren KV-Leistungen bis zur vollständigen elektronischen Akte mit zusätzlichen ERP-Funktionen reicht. In dieser Software (hier im Folgenden mit PVS bezeichnet) steht normalerweise die Funktion des STDT-Exports zur Verfügung. Der STDT ist ein Mitglied der Datenstandard-Familie, die von der KBV als xDT-Familie betreut wird. Dieser sogenannte Statistik-Datenträger enthält in strukturierter Form alle für das geplante Modell erforderlichen Daten in strukturierter Form und noch einige weitere darüber hinaus.

2.2 Datenseparation

Der STDT sieht vor, dass die einzelnen Patienten (beziehungsweise Namen und Vornamen) durch numerische Indizes in austeigender Reihenfolge repräsentiert werden und unter dem entsprechenden Index alle im Beobachtungs- oder Exportzeitkorridor auftretenden Ereignisse zu dem entsprechenden Patienten subsummiert werden. Zur Generierung dieser Indizes werden üblicherweise die (numerischen Primär-)Schlüssel der dem PVS zugrunde liegenden Datenbank benutzt. Diese sind allerdings bei den PVSen der einzelnen Teilnehmer an der Datensammlung im Allgemeinen zu jedem Patienten disjunkt. Um die Daten über das Pseudonym zusammenführen zu können, ist deshalb die Abbildung dieser Nummer mit einem in allen Systemen identischen, identifizierenden Datum erforderlich. Dazu wurde im vorliegenden Modell entweder die neue, von der eGK abgeleitete Versichertennummer des Patienten oder die Kombination Versichertennummer mit Krankenkassennummer (IK) bei Patienten mit der alten KVK gewählt. Diese, zu jedem Patienten generierten Datenpaare aus Identifikator und Nummer, werden vom PVS ebenfalls in eine Datei mit „Identifikationsdaten“ (Dateiendung .IDAT) exportiert. Sollte bei einem Patienten ein Übergang von KVK auf eGK im Lauf des Modellversuchs erfolgen, so ist vorgesehen, dass zumindest einmal das Datentripel von eGK-Versichertennummer, KVK-Versichertennummer mit Krankenkassennummer und Ordnungsnummer des PVS für eine Datenlieferung genutzt wird, so dass die Pseudonyme aus der eGK-Versichertennummer mit dem Pseudonym aus KVK-Versichertennummer und Krankenkassennummer zugeordnet werden können und damit kein Bruch in der pseudonymen Datenhaltung eintritt.

Um die für die Auswertung nicht benötigten Daten bereits frühestmöglich auf das erforderliche Maß zu reduzieren, werden die Daten nach dem Export an Hand der Filterdatei (Namensvorgabe: „Ersetzungstabelle.reg“) um alle Einträge, die nicht aufgeführt oder als „remove“ gekennzeichnet sind, bereinigt. Die mit „keep“ beschriebenen Einträge werden unverändert übernommen, die als „modify“ beschriebenen Einträge werden verändert. Im gleichen Durchgang werden alle Ordnungsnummern in den personengebundenen Datensätzen der Datei, die zu einer lokalen Rück-Zuordnung führen könnten, durch eine bei jedem Exportvorgang neu zu generierende UID ersetzt. Die entstehende Datei (Dateiendung .MDAT) enthält somit „vorpseudonymisierte“ medizinische Daten.

Eine Liste der genutzten und zu verändernden Einträge wird im Fachkonzept Controlling beschrieben.

Außerdem werden aus dem STDT-Datensatz die Daten des Abschnitts „besa“, der konkrete Informationen zu der die Daten liefernden Praxis enthält, extrahiert. Statt dessen wird an Stelle der „besa“-Informationen eine Einmal-UID in den Datensatz eingefügt und aus der BSNR (Betriebstättennummer), die Inhalt des „besa“ ist, eine eigene, den „Identifikationsdaten“ der Patienten entsprechende Datei (Dateiendung .BDAT) für den Datenlieferanten erzeugt.

Die so erzeugten Dateien (*.MDAT, *.BDAT, *.IDAT) stellen die Eingangsdaten des nun folgenden kryptographischen Pseudonymisierungsvorganges dar.

2.3 Pseudonymisierung

Die Pseudonymisierung besteht aus drei Schritten, von denen jeder die Qualität des „Pseudonyms“ verbessert. Der „nullte“ Schritt ist bereits im vorhergehenden Absatz beschrieben, ein Teil der Daten wird entfernt, ein anderer verändert. Die verbleibenden Daten werden bereits in der Arztpraxis einer „Vor“-Pseudonymisierung unterzogen, die es ohne Hilfe des teilnehmenden Arztes bereits nach dieser Stufe nahezu unmöglich macht, einzelne Patienten zu repersonalisieren.

Die Voraussetzung für das gesamte Verfahren, das auf mathematischen Prinzipien der „Primen Restgruppen“ basiert und für dessen kryptographische Operationen sich die Funktionen der openSSL-Bibliothek eignen, ist als Basis ein Erzeuger einer Primen Restklasse und eine hinreichend große geeignete Primzahl erforderlich:

Erzeuger der primen Restklasse: *g, p*

Zusätzlich müssen für die weitere Verarbeitung zwei Schlüsselpaare erzeugt werden. Auf Grund der mathematischen Basis des Verfahrens sind dazu Schlüsselpaare nach Diffie-Hellman (DH-Verfahren) zu wählen:

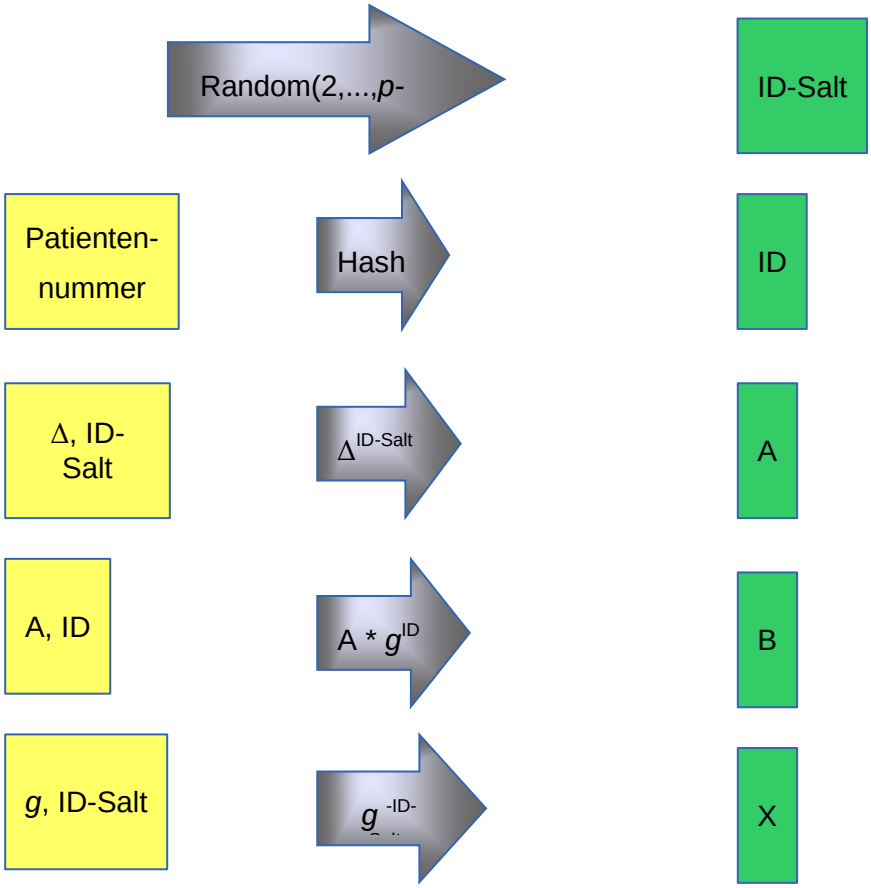
Diffie-Hellmann-Schlüsselpaar der TTP: ε (privat), E (öffentlich)

Vom Schlüsselpaar der TTP wird nur der private Schlüssel verwendet. Der öffentliche Schlüssel entsteht im Verfahren einfach mit. Da die beiden Schlüssel funktional symmetrisch sind, könnte jeder der beiden Schlüssel jeweils beide Funktionen ausfüllen. Das zweite Schlüsselpaar wird der Auswertestelle zugeordnet:

Diffie-Hellmann-Schlüsselpaar der AWS: δ (privat), Δ (öffentlich)

Die Schlüssel dieses Schlüsselpaares werden beide genutzt. Auf Basis und mit Hilfe dieser kryptographischen Objekte kann die Pseudonymisierung stattfinden.

Die erste Station ist die Praxis des die Daten erzeugenden Arztes. Hier findet bereits eine Vorpseudonymisierung statt, die es einem Angreifer, der z.B. bereits die TTP „gekapert“ hätte, sehr schwer machen würde, die empfangenen Daten auf einen konkreten Patienten herunter zu brechen. Im ersten Schritt wird dazu für jeden Versicherten die beschriebene eindeutige Identifikationsinformation, also entweder die Versichertennummer der eGK oder die Nummernkombination der KVK aus Versichertennummer und Krankenkassennummer, einem Hash-Verfahren unterzogen. Danach wird eine hinreichend große Zufallszahl erzeugt (als Salt) und aus diesem Salt und dem öffentlichen Schlüssel der Auswertestelle wird ein weiterer temporärer Schlüssel (A) erzeugt. Dieser Schlüssel ist für jeden Patienten und bei jeder einzelnen Datenlieferung unterschiedlich.



Mit Hilfe des „Schlüssels“ A wird aus der ge“hash“ten Patientennummer das erste von zwei zu übertragenden kryptographischen Objekten B erzeugt. Das zweite zu übertragende Objekt (X) wird aus dem Erzeuger der Primen Restgruppe und der erzeugten Zufallszahl gebildet. Dieses Verfahren wird für alle Patienten und für die Daten der liefernden Praxis (Inhalt der .BDAT-Datei) durchgeführt.

Danach werden die medizinischen Daten, bei denen der Patientenbezug nun verwischt ist, mit dem öffentlichen D2D-Schlüssel der Auswertestelle verschlüsselt (erhält die Dateiendung .XXX), so dass die nächste Station, die TTP, diese Daten nicht einsehen kann.

Alle drei Dateien werden dann verschlüsselt an die TTP übertragen.

Bei der TTP (Trusted Third Party, Pseudonymisierungsstelle) werden zwei der drei Dateien entschlüsselt. Die Datei mit den medizinischen Inhalten (Endung .XXX bzw. .MDAT.XXX) kann bei der TTP nicht entschlüsselt werden, da sie mit dem priva-

ten Schlüssel der Auswertestelle verschlüsselt ist. Bis zur Weiterleitung der Ergebnisse der Verarbeitung der weiteren Daten wird sie zwischengespeichert.

Die Inhalte der beiden anderen Dateien bestehen aus Tupeln,

die bei der Datei *.IDAT die folgende Form hat:

UUID_1	B_{eGK}_1	B_{KVK}_1	X_1
UUID_2	B_{eGK}_2	B_{KVK}_2	X_2
UUID_3	B_{eGK}_3	B_{KVK}_3	X_3
UUID_4	B_{eGK}_4	B_{KVK}_4	X_4
UUID_5	B_{eGK}_5	B_{KVK}_5	X_5
…	…	…	...

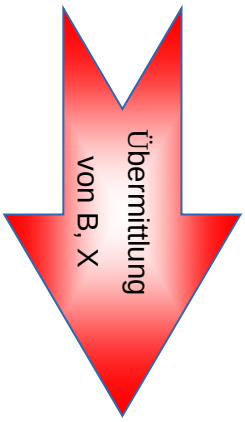
wobei jedes dieser Tupel einem Patienten zugeordnet ist. Eine der beiden Werte BeGK_x bzw. BKVK_x kann leer sein.

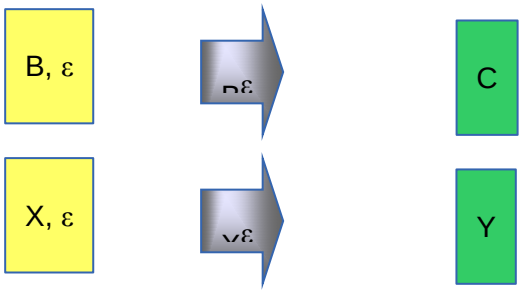
Die Datei mit der Endung .BDAT enthält nur ein Tupel:

$UUID_{BSNR}$ B_{BSNR} X_{BSNR}

da bei der Praxis (BSNR) keine zwei Identifikatoren vorgesehen werden müssen.

Die Werte B und X sind die Eingangsvariablen der folgenden Operationen, die Parameter der Operationen sind wieder der Erzeuger der Primen Restgruppe, die zugehörige Primzahl und der private Schlüssel der TTP:

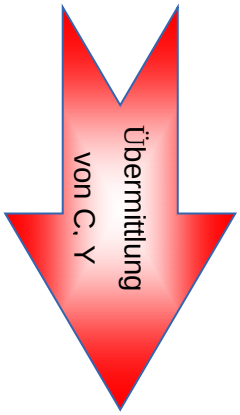


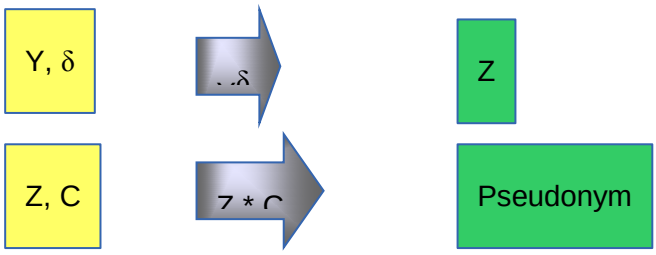


Aus den Werten B entsteht mit Hilfe der Potenzierung nach den algebraischen Methoden der Primen Restklassen die Werte C und aus den Werten X entstehen auf die gleiche Weise die Y-Werte. In den beiden Dateien werden die B-Werte durch die korrespondierenden C-Werte ersetzt, die X-Werte durch die korrespondierenden Y-Werte.

Nach dieser Intermediär-Pseudonymisierung werden die entstandenen beiden Dateien zusammen mit der zwischengespeicherten .XXX-Datei verschlüsselt an die Auswertestellen versandt.

Mit den Werten C und Y arbeitet die Auswertestelle weiter. Sie stellen noch immer nicht das eigentliche Pseudonym dar und sind für jeden Patienten bei jeder Übertragung unterschiedlich und per se nicht zusammenführbar. Die Parameter der Operationen sind wieder der Erzeuger der Primen Restgruppe, die zugehörige Primzahl und der private Schlüssel der AWS (Auswertestelle):





Aus dem Eingangswert Y und dem privaten Schlüssel der AWS entsteht durch Potenzierung der Zwischenwert Z. Das Produkt aus Z und dem anderen Eingangswert C entsteht nun das eigentliche Pseudonym, das für jeden Patienten (mit überwältigender Wahrscheinlichkeit) eindeutig ist.

Dieses Pseudonym ist über das entsprechende Tupel mit einer UID verbunden. Da die AWS mit ihrem privaten Schlüssel die .XXX-Datei in eine .MDAT-Datei entschlüsseln kann, ist es möglich, die medizinischen Daten, die in dieser Datei ebenfalls an der entsprechenden UID hängen, jeweils einem Pseudonym zuzuordnen. Die in gleicher Art bearbeiteten Daten der .BDAT-Datei liefern das Pseudonym der liefernden Praxis, so dass die pseudonymen Patientendaten jeweils eindeutig einem Praxispseudonym zugeordnet werden können. Mit diesen Daten kann nun eine Datenbank beschickt werden.

2.4 Auswertung

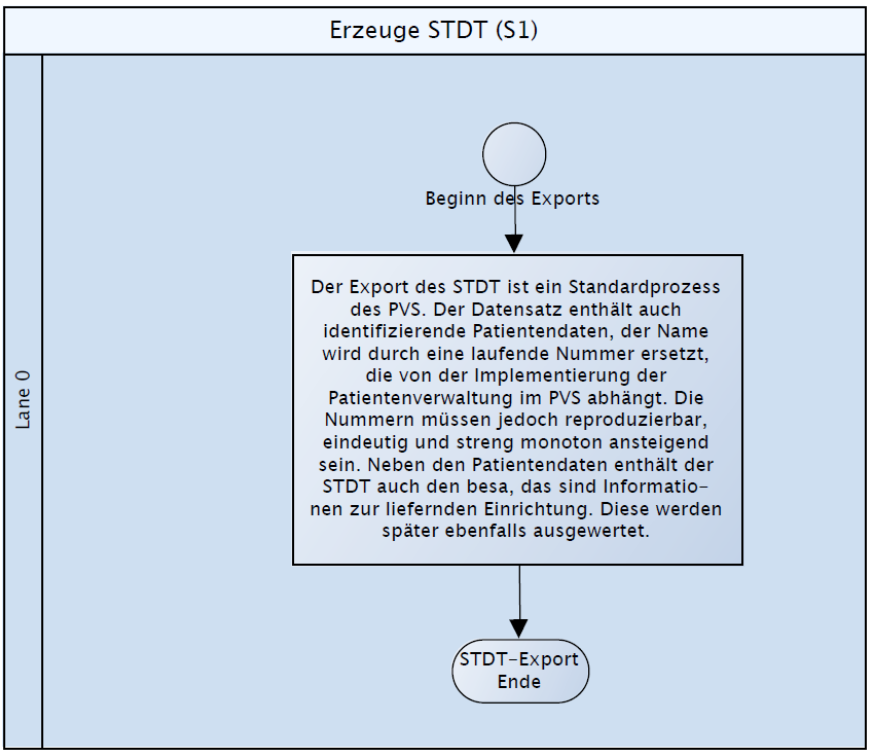
Für die Auswertung der pseudonymen Daten steht somit eine stetig wachsende Basis zur Verfügung. Die Auswerteziele und die Kriterien legt die KV Nordrhein im Auftrag und nach Absprache mit den Ärzten der DAGIV fest. Limitierender Faktor jeder Auswertung ist die Wahrung der k-Anonymität der Patienten. Diese muss für jede Fragestellung gesondert untersucht und berücksichtigt werden.

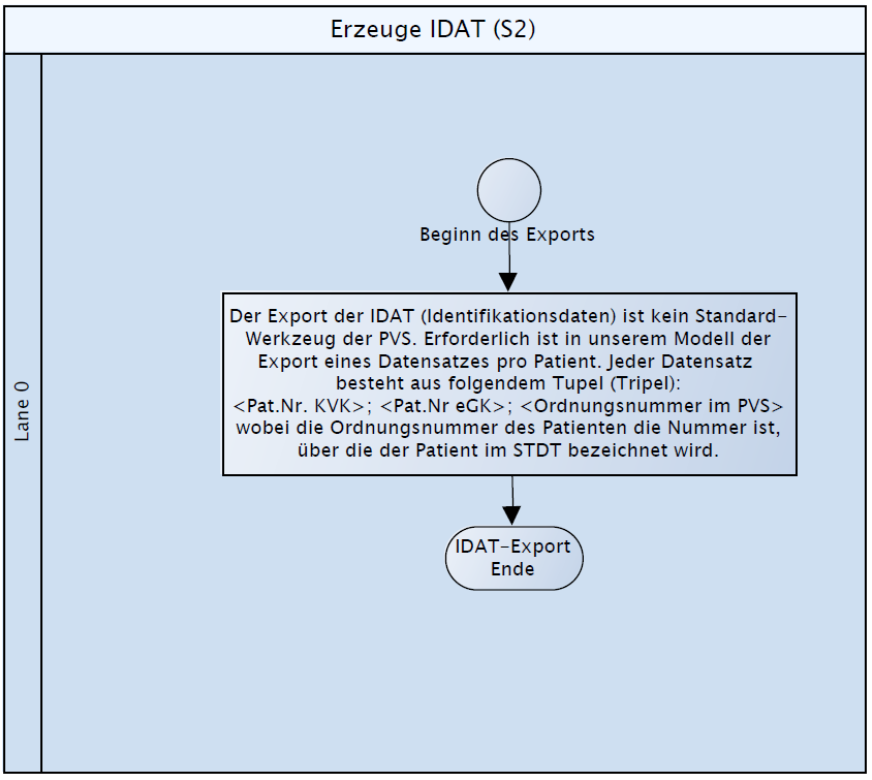
Für weitere Ausführungen zur Auswertung wird auf die Dokumentation der AWS verwiesen.

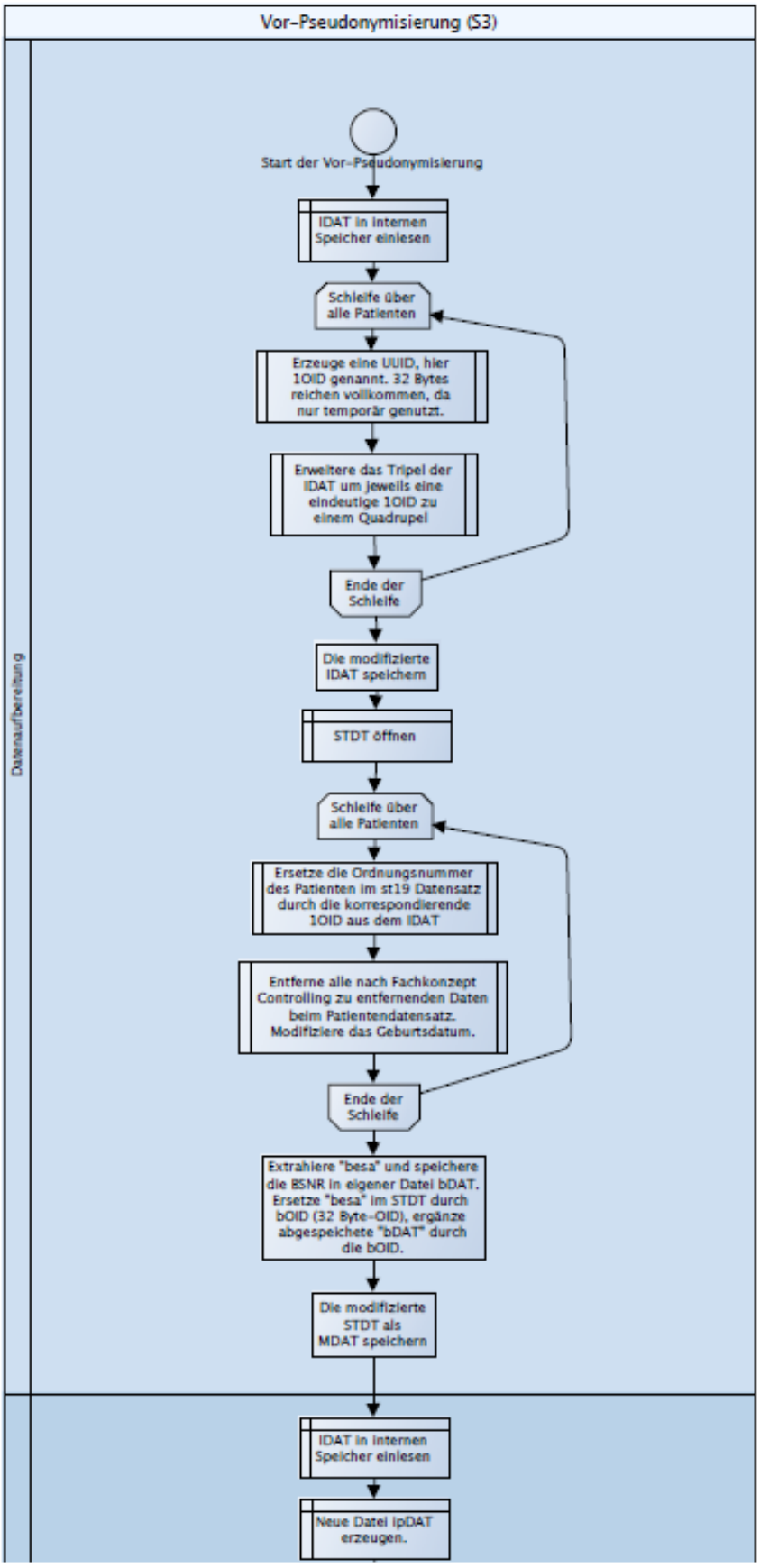
2.5 Mathematisches Prinzip

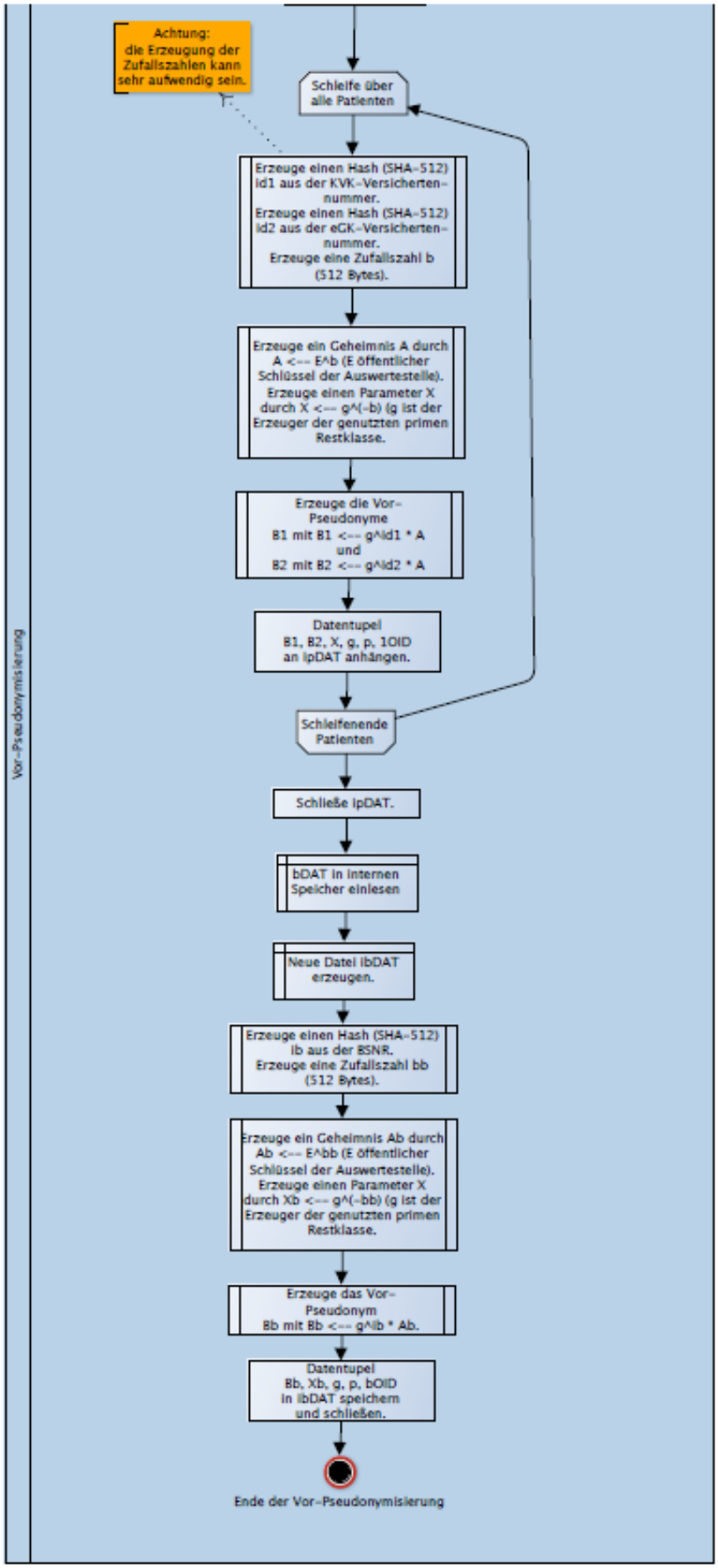
Die Pseudonymisierung in der vorliegenden Form beruht auf den Eigenschaften von Primen Restgruppen.

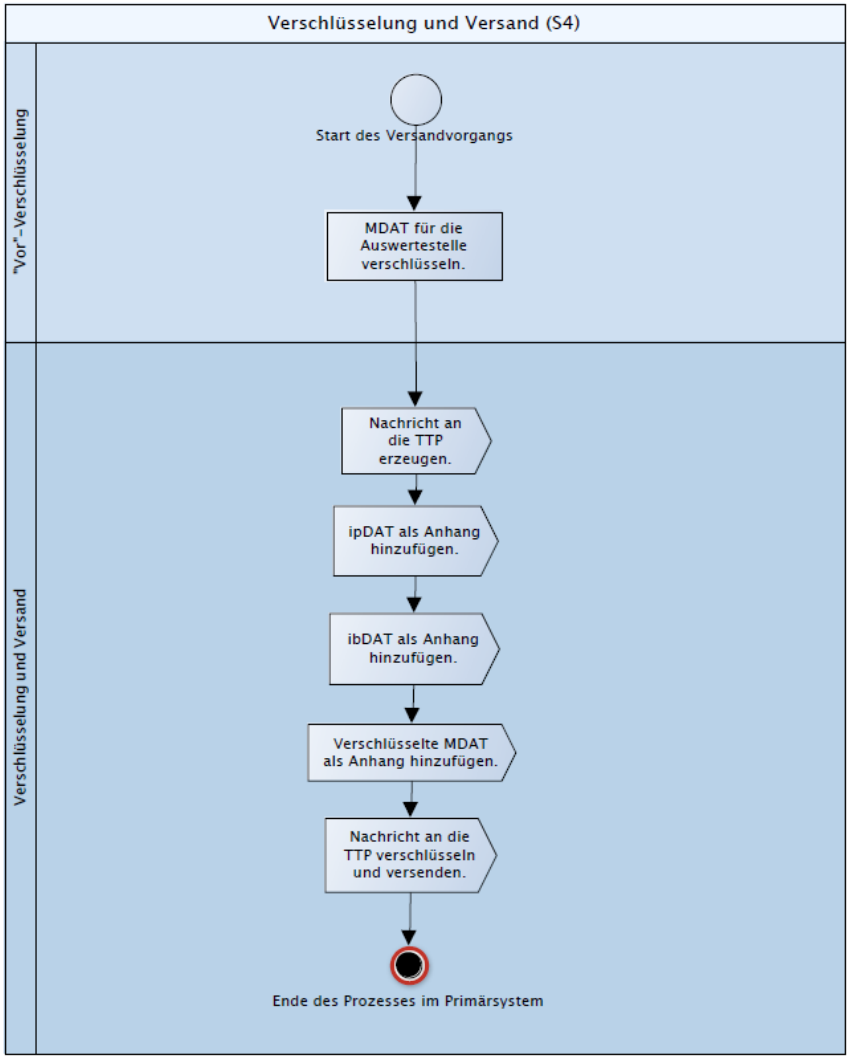
3 Grafiken

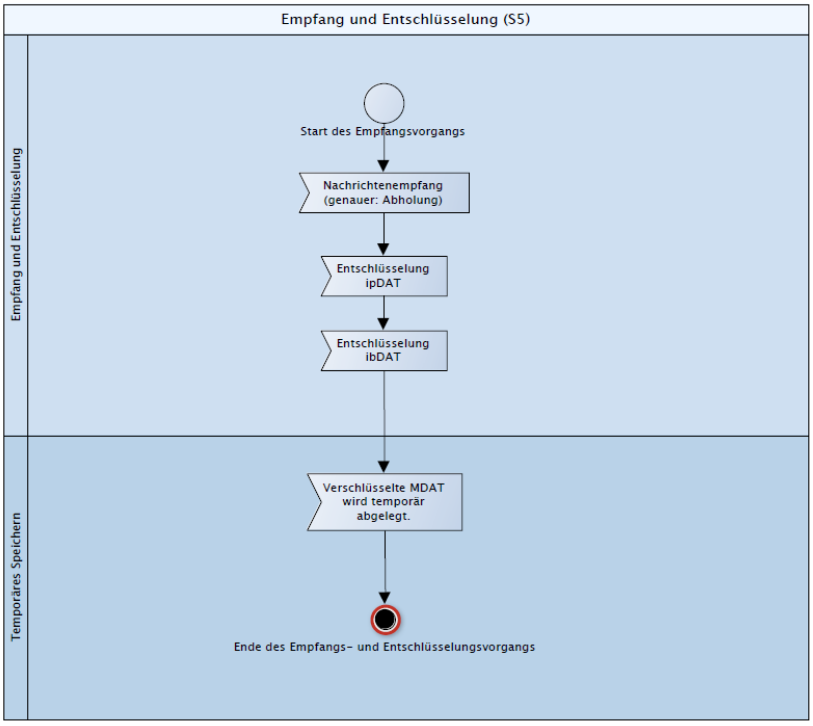

Erzeuge STDT (S1)
Lane 0
Beginn des Exports
Der Export des STDT ist ein Standardprozess des PVS. Der Datensatz enthält auch identifizierende Patientendaten, der Name wird durch eine laufende Nummer ersetzt, die von der Implementierung der Patientenverwaltung im PVS abhängt. Die Nummern müssen jedoch reproduzierbar, eindeutig und streng monoton ansteigend sein. Neben den Patientendaten enthält der STDT auch den besa, das sind Informationen zur liefernden Einrichtung. Diese werden später ebenfalls ausgewertet.
STDT-Export Ende

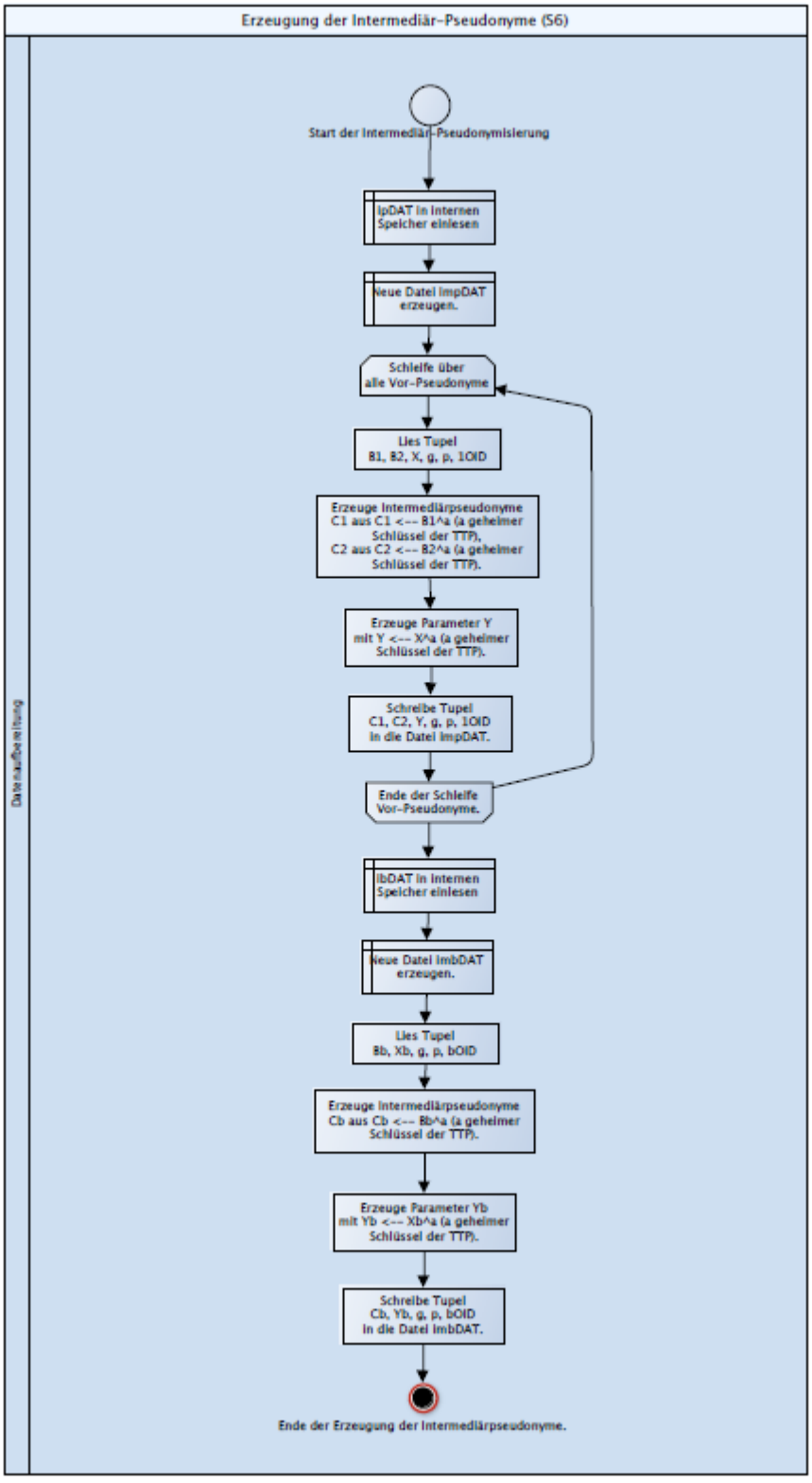

Erzeuge IDAT (S2)
Lane 0
Beginn des Exports
Der Export der IDAT (Identifikationsdaten) ist kein Standard-Werkzeug der PVS. Erforderlich ist in unserem Modell der Export eines Datensatzes pro Patient. Jeder Datensatz besteht aus folgendem Tupel (Tripel):
<Pat.Nr. KVK>; <Pat.Nr eGK>; <Ordnungsnummer im PVS>
wobei die Ordnungsnummer des Patienten die Nummer ist, über die der Patient im STDT bezeichnet wird.
IDAT-Export Ende

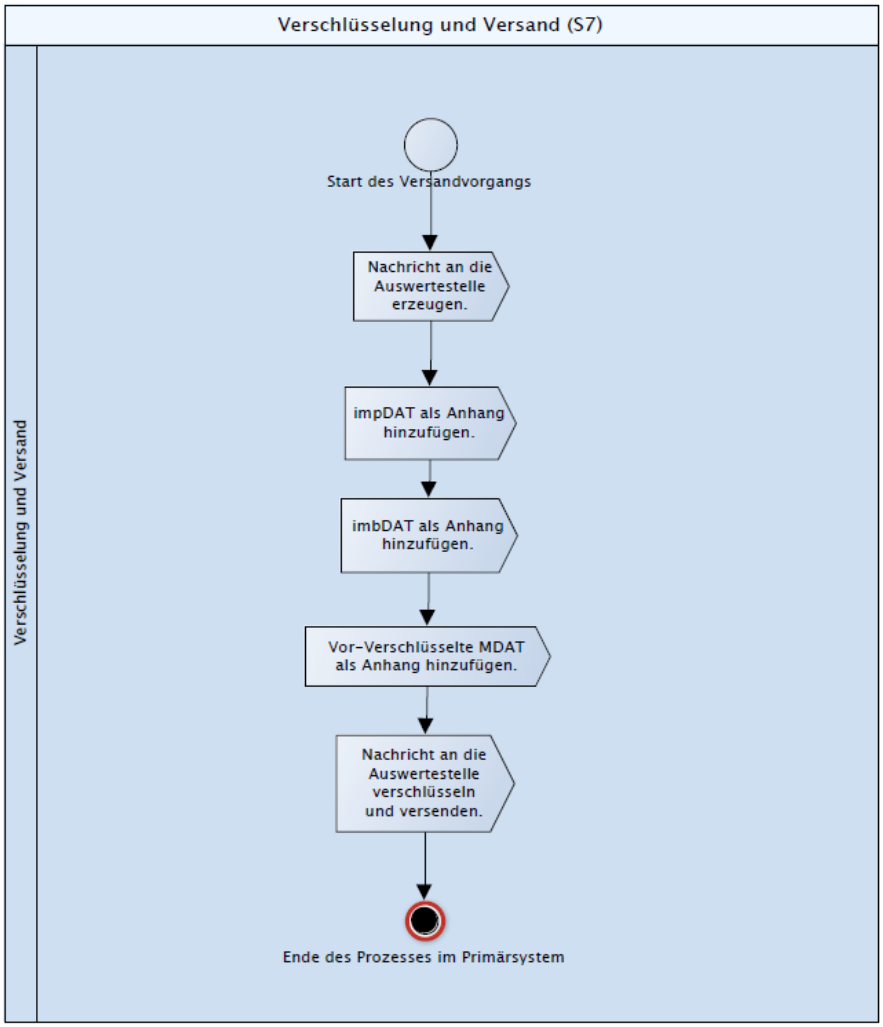

Vor-Pseudonymisierung (S3)
Datenaufbereitung
Start der Vor-Pseudonymisierung
IDAT in internen Speicher einlesen
Schleife über alle Patienten
Erzeuge eine UUID, hier 1OID genannt. 32 Bytes reichen vollkommen, da nur temporär genutzt.
Erweitere das Tripel der IDAT um jeweils eine eindeutige 1OID zu einem Quadrupel
Ende der Schleife
Die modifizierte IDAT speichern
STDT öffnen
Schleife über alle Patienten
Ersetze die Ordnungsnummer des Patienten im st19 Datensatz durch die korrespondierende 1OID aus dem IDAT
Entferne alle nach Fachkonzept Controlling zu entfernenden Daten beim Patientendatensatz. Modifiziere das Geburtsdatum.
Ende der Schleife
Extrahiere "besa" und speichere die BSNR in eigener Datei bDAT. Ersetze "besa" im STDT durch bOID (32 Byte-OID), ergänze abgespeichete "bDAT" durch die bOID.
Die modifizierte STDT als MDAT speichern
IDAT in internen Speicher einlesen
Neue Datei ipDAT erzeugen.

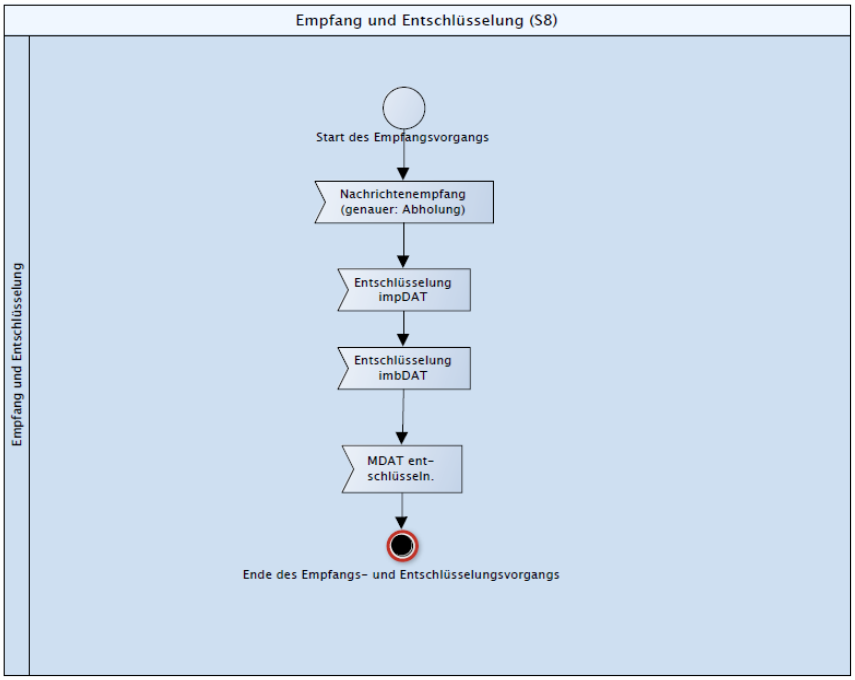

Vor-Pseudonymisierung
Achtung:
die Erzeugung der Zufallszahlen kann sehr aufwendig sein.
Schleife über alle Patienten
Erzeuge einen Hash (SHA-512) id1 aus der KVK-Versicherten-nummer.
Erzeuge einen Hash (SHA-512) id2 aus der eGK-Versicherten-nummer.
Erzeuge eine Zufallszahl b (512 Bytes).
Erzeuge ein Geheimnis A durch A <-- E^b (E öffentlicher Schlüssel der Auswertestelle).
Erzeuge einen Parameter X durch X <-- g^(-b) (g ist der Erzeuger der genutzten primen Restklasse.
Erzeuge die Vor-Pseudonyme
B1 mit B1 <-- g^id1 * A
und
B2 mit B2 <-- g^id2 * A
Datentupel B1, B2, X, g, p, 1OID an ipDAT anhängen.
Schleifenende Patienten
Schließe ipDAT.
bDAT in internen Speicher einlesen
Neue Datei ibDAT erzeugen.
Erzeuge einen Hash (SHA-512) ib aus der BSNR.
Erzeuge eine Zufallszahl bb (512 Bytes).
Erzeuge ein Geheimnis Ab durch Ab <-- E^bb (E öffentlicher Schlüssel der Auswertestelle).
Erzeuge einen Parameter X durch Xb <-- g^(-bb) (g ist der Erzeuger der genutzten primen Restklasse.
Erzeuge das Vor-Pseudonym
Bb mit Bb <-- g^ib * Ab.
Datentupel Bb, Xb, g, p, bOID in ibDAT speichern und schließen.
Ende der Vor-Pseudonymisierung

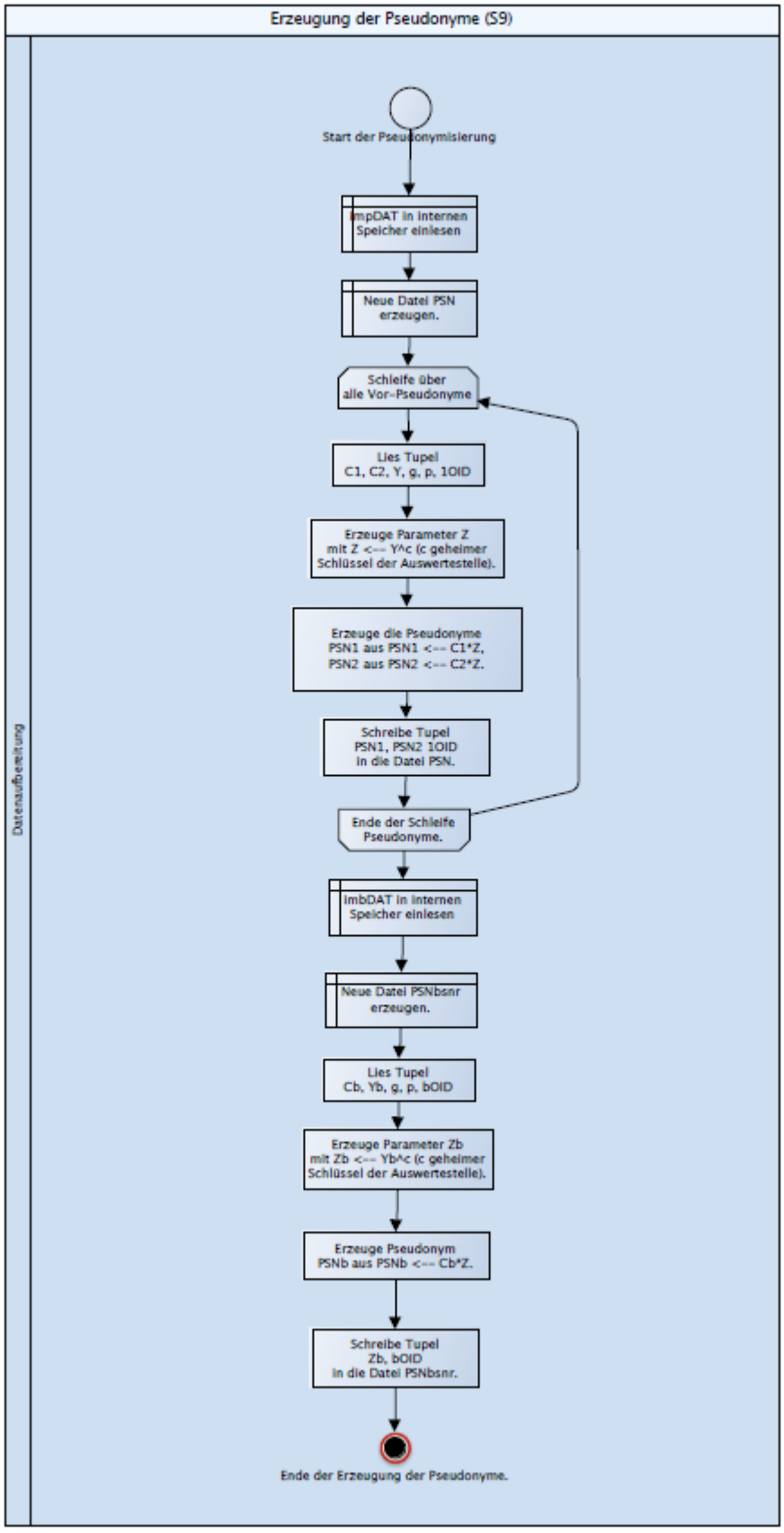

Verschlüsselung und Versand (S4)
"Vor"-Verschlüsselung
Start des Versandvorgangs
MDAT für die Auswertestelle verschlüsseln.
Verschlüsselung und Versand
Nachricht an die TTP erzeugen.
ipDAT als Anhang hinzufügen.
ibDAT als Anhang hinzufügen.
Verschlüsselte MDAT als Anhang hinzufügen.
Nachricht an die TTP verschlüsseln und versenden.
Ende des Prozesses im Primärsystem


Empfang und Entschlüsselung (S5)
Empfang und Entschlüsselung
Temporäres Speichern
Start des Empfangsvorgangs
Nachrichtenempfang (genauer: Abholung)
Entschlüsselung ipDAT
Entschlüsselung ibDAT
Verschlüsselte MDAT wird temporär abgelegt.
Ende des Empfangs- und Entschlüsselungsvorgangs


Erzeugung der Intermediär-Pseudonyme (S6)
Datenaufbereitung
Start der Intermediär-Pseudonymisierung
IpDAT in internen Speicher einlesen
Neue Datei ImpDAT erzeugen.
Schleife über alle Vor-Pseudonyme
Lies Tupel B1, B2, X, g, p, 1OID
Erzeuge Intermediärpseudonyme C1 aus C1 <-- B1^a (a geheimer Schlüssel der TTP), C2 aus C2 <-- B2^a (a geheimer Schlüssel der TTP).
Erzeuge Parameter Y mit Y <-- X^a (a geheimer Schlüssel der TTP).
Schreibe Tupel C1, C2, Y, g, p, 1OID in die Datei ImpDAT.
Ende der Schleife Vor-Pseudonyme.
IbDAT in internen Speicher einlesen
Neue Datei ImbDAT erzeugen.
Lies Tupel Bb, Xb, g, p, bOID
Erzeuge Intermediärpseudonyme Cb aus Cb <-- Bb^a (a geheimer Schlüssel der TTP).
Erzeuge Parameter Yb mit Yb <-- Xb^a (a geheimer Schlüssel der TTP).
Schreibe Tupel Cb, Yb, g, p, bOID in die Datei ImbDAT.
Ende der Erzeugung der Intermediärpseudonyme.


Verschlüsselung und Versand (S7)
Verschlüsselung und Versand
Start des Versandvorgangs
Nachricht an die Auswertestelle erzeugen.
impDAT als Anhang hinzufügen.
imbDAT als Anhang hinzufügen.
Vor-Verschlüsselte MDAT als Anhang hinzufügen.
Nachricht an die Auswertestelle verschlüsseln und versenden.
Ende des Prozesses im Primärsystem


Empfang und Entschlüsselung (S8)
Empfang und Entschlüsselung
Start des Empfangsvorgangs
Nachrichtenempfang (genauer: Abholung)
Entschlüsselung impDAT
Entschlüsselung imbDAT
MDAT ent-schlüsseln.
Ende des Empfangs- und Entschlüsselungsvorgangs


Erzeugung der Pseudonyme (S9)
Datenaufbereitung
Start der Pseudonymisierung
impDAT in internen Speicher einlesen
Neue Datei PSN erzeugen.
Schleife über alle Vor-Pseudonyme
Lies Tupel C1, C2, Y, g, p, 1OID
Erzeuge Parameter Z mit Z <-- Y^c (c geheimer Schlüssel der Auswertestelle).
Erzeuge die Pseudonyme PSN1 aus PSN1 <-- C1*Z, PSN2 aus PSN2 <-- C2*Z.
Schreibe Tupel PSN1, PSN2 1OID in die Datei PSN.
Ende der Schleife Pseudonyme.
imbDAT in internen Speicher einlesen
Neue Datei PSNbsnr erzeugen.
Lies Tupel Cb, Yb, g, p, bOID
Erzeuge Parameter Zb mit Zb <-- Yb^c (c geheimer Schlüssel der Auswertestelle).
Erzeuge Pseudonym PSNb aus PSNb <-- Cb*Z.
Schreibe Tupel Zb, bOID in die Datei PSNbsnr.
Ende der Erzeugung der Pseudonyme.

4 Openssh-Script

```
REM Batch-Datei (Windows) zur Erzeugung der Diffie-Hellmann-Schluessel für die

REM TTP und die Auswertestelle der DAGIV-Pseudonymisierungsstelle

REM vor der Verteilung muss aus dem TTP-Keystore noch der private Schlüssel entfernt

REM werden (klappt mit Text-Editor)


REM #date

REM Erzeugt einen Parametersatz (einen Erzeuger) für eine prime Restgruppe

REM REM C:\OpenSSL-Win32\bin\openssl.exe dhparam -2 -outform PEM -out dhp.pem -C 1024
-text

REM #date

REM #oder einfacher ?!? mit einem Generator != 2 und einem Parameter Q

REM             : openssl dsaparam -out dsaparam.pem 1024 -text

REM #oder          : openssl dsaparam -out dsaparam.pem 2048 -text

REM #oder mit Zusatzoutput:

REM #         openssl dhparam -in dsaparam.pem -text !!!

REM #         openssl dhparam -in dsaparam.pem -C   ist mit C-Code

REM #zeige die Parameter

REM REM C:\OpenSSL-Win32\bin\openssl.exe dhparam -inform PEM -in dhp.pem -text -C

REM # neue Parameterbildung:

C:\OpenSSL-Win32\bin\openssl.exe genpkey -genparam -algorithm DH -out dhp.pem -text -
pkeyopt dh_paramgen_prime_len:1024


REM # generiert private und public key

C:\OpenSSL-Win32\bin\openssl.exe genpkey -paramfile dhp.pem -out keysAWS.pem -text


REM # schmeisst private und public key

C:\OpenSSL-Win32\bin\openssl.exe genpkey -paramfile dhp.pem -out keysTTP.pem -text
```