

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro



(43) Internationales Veröffentlichungsdatum
3. September 2009 (03.09.2009)

(10) Internationale Veröffentlichungsnummer
WO 2009/106055 A2

(51) Internationale Patentklassifikation:
H04L 9/06 (2006.01)

(21) Internationales Aktenzeichen: PCT/DE2009/000258

(22) Internationales Anmeldedatum:
21. Februar 2009 (21.02.2009)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
10 2008 011 882.6
29. Februar 2008 (29.02.2008) DE

(71) Anmelder und

(72) Erfinder: **NIGGL, Robert** [DE/DE]; Untersbergstrasse
30, 83435 Bad Reichenhall (DE).

(74) Anwalt: **SÖFFGE & SÖFFGE . BERG**; Moltkestrasse
5, 80803 München (DE).

(81) Bestimmungsstaaten (soweit nicht anders angegeben, für
jede verfügbare nationale Schutzrechtsart): AE, AG, AL,
AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY,
BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO,

DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP,
KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD,
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE,
SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ,
UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Bestimmungsstaaten (soweit nicht anders angegeben, für
jede verfügbare regionale Schutzrechtsart): ARIPO (BW,
GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG,
ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, MD, RU,
TJ, TM), europäisches (AT, BE, BG, CH, CY, CZ, DE,
DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT,
LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI,
SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, ML, MR, NE, SN, TD, TG).

Veröffentlicht:

— ohne internationalen Recherchenbericht und erneut zu
veröffentlichen nach Erhalt des Berichts (Regel 48 Absatz
2 Buchstabe g)

(54) Title: SYSTEM FOR THE CONTROLLED DATA EXCHANGE BETWEEN AT LEAST TWO DATA CARRIERS VIA
MOBILE READ-WRITE MEMORIES

(54) Bezeichnung: SYSTEM ZUM KONTROLLIERTEN DATENAUSTAUSCH ZWISCHEN MINDESTENS ZWEI DATEN-
TRÄGERN ÜBER MOBILE SCHREIB-LESE-SPEICHER

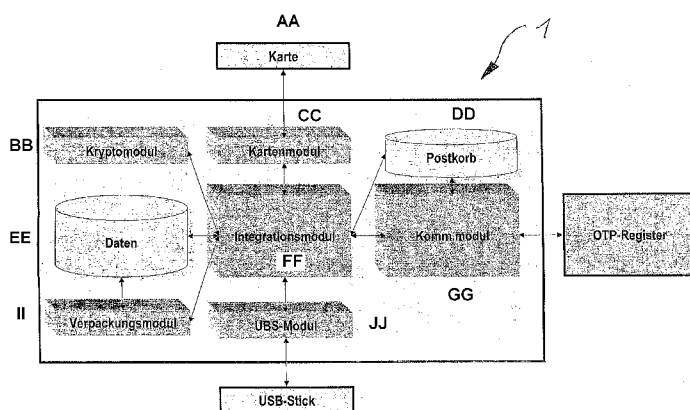


Fig. 1

AA ... Card
BB ... Cryptographic module
EE ... Data
II ... Packing module
CC ... Card module
FF ... Integration module
JJ ... UBS module
DD ... In-basket
GG ... Communication module

(57) Abstract: The present invention relates to a data transmission system (1), which enables a controlled data exchange between at least two data carriers via mobile (location-independent) read-write memories, particularly a system for the secure forwarding of individual data to predetermined recipients while controlling those involved. Those involved, or the mobile memories, do not have to be identified/registered in any way with respect to the system, which is to say they can remain completely anonymous. The respective output data, such as a patient file or a fingerprint, are represented as ONE-TIME-PAD key cipher pairs, and the pair components are always distributed among different mobile and central intermediate storage devices.

(57) Zusammenfassung: Mit der vorliegenden Erfindung wird ein Daten-Übertragungs-System (1) vorgestellt, das einen kontrollierten Datenaustausch zwischen mindestens zwei Datenträgern über mobile (ortsunabhängige) Schreib-Lese-Speicher ermöglicht, insbesondere ein System zur sicheren Weiterleitung individueller Daten an vorbestimmte Empfänger unter Kontrolle der

[Fortsetzung auf der nächsten Seite]



Betroffenen. Die Betroffenen bzw. die mobilen Speicher selbst müssen dazu in keiner Weise systembezogen identifiziert/registriert werden, d.h. können vollständig anonym bleiben. Die jeweiligen Ausgangsdaten, z.B. eine Patientenakte oder ein Fingerprint, werden als ONE-TIME-PAD-Schlüssel-Chiffre-Paar dargestellt und die Paar-Komponenten immer auf verschiedene mobile und zentrale Zwischenspeicher verteilt.

SYSTEM ZUM KONTROLLIERTEN DATENAUSTAUSCH ZWISCHEN MINDESTENS ZWEI DATENTRÄGERN ÜBER MOBILE SCHREIB-LESE-SPEICHER

Die vorliegende Erfindung befasst sich mit einer Vorrichtung und einem Verfahren zum kontrollierten Datenaustausch zwischen mindestens zwei Datenträgern über mobile Schreib-Lese-Speicher, insbesondere mit einem System zur Aufnahme und Aufbereitung individueller Daten zur sicheren Weiterleitung an vorbestimmte Empfänger.

Derartige Verfahren und Systeme sind im Stand der Technik bekannt. Die so transportierten Daten weisen in der Regel den Mangel auf, mit geeigneten Mitteln aufgebrochen und gefälscht zu werden, sodass sie im Ergebnis nicht sicher genug sind, insbesondere nicht zukunftssicher im Hinblick auf zukünftige Rechensysteme mit entsprechender Kapazität (Stichwort: Quantencomputer). Beispielsweise ist aus der WO 2007/109373 A2 ein Verfahren zur Komprimierung von Datenmengen bei modernen Verschlüsselungsmethoden bekannt geworden, worauf nachfolgend im jeweiligen Kontext Bezug genommen wird.

Daher ist es Aufgabe der vorliegenden Erfindung, ein System, das nach einem bestimmten Verfahren arbeitet, bereitzustellen, das in der Lage ist, individuelle Daten sicher aufzunehmen, zu verarbeiten und formgerecht und sicher zu oder von einem vorbestimmten Datenträger weiter zu leitet.

Diese Aufgabe wird mit den kennzeichnenden Merkmalen der Hauptansprüche gelöst.

Erfindungsgemäß ist das Verfahren zum kontrollierten Datenaustausch zwischen Datenträgersystemen mit mobilen (d.h. ortsunabhängigen) Schreib-Lese-Speichern dadurch gekennzeichnet, dass ein erstes Trägersystem (**Quellsystem**) beliebige serielle Ausgangsdaten (S) via Kryptomodul als **ONE-TIME-PAD-Schlüssel-Chiffre-Paar** darstellt und dann via Integrationsmodul extern zwischenspeichert, wobei immer **eine Paar-Komponente (K₁)** abrufbar -- also insbesondere mit systemweit geeigneter Referenz -- via DFÜ auf einem zentralen Zwischenspeicher abgelegt wird und die **andere Paar-Komponente (K₂)** und die K₁-Referenz auf mobilen Speichern, so dass ein zweites Trägersystem (**Zielsystem**) durch Anschluss und Auswertung der mobilen Speicher und durch Abruf der zentral hinterlegten Komponente zunächst das OTP-Schlüssel-Chiffre-Paar und schließlich durch Entschlüsselung die Ausgangsdaten selbst gewinnen kann.

Das erfindungsgemäße Trägersystem ist also insbesondere dadurch gekennzeichnet, dass es mit einem OTP-Kryptomodul ausgestattet ist und

- (a) als Quellsystem beliebige serielle Ausgangsdaten (S) in einer vorbestimmten Form von OTP-Schlüssel-Chiffre-Paaren kryptonisiert (s. unten); und
- (b) als Zielsystem aus OTP-Schlüssel-Chiffre-Paaren Ausgangsdaten rekonstruiert.

Eine OTP- Datenkomprimierung, z.B. nach WO 2007/109373 A2, wird nicht vorausgesetzt. Die OTP-Kryptoeinheit eines Quellsystems erzeugt also randomisiert einen Einmalschlüssel E, der die gleiche Länge wie S hat und verschlüsselt S mit E über eine Bit-Verknüpfung (Chiffre V).

Beispielhaft sei als Bit-Verknüpfung die Bit-Addition genannt: $0+1=1+0=1$, $1+1=0+0=0$.

Dann ist $V=S+E$ und $S=V+E$, wobei jeweils bitweise addiert wird.

Anmerkung zur mathematischen Literatur: Die Bit-Addition hat die Eigenschaften einer „abelschen Gruppe“; man kann also z.B. rechnen „wie mit ganzen Zahlen“. Die Bit-Addition ist in der Algebra auch bekannt als „Addition im kleinsten Körper ($\mathbb{F}_2$)“.

Bisher liefern nur die im Rahmen der Quantenkryptographie entwickelten technischen Verfahren zur Schlüsselgenerierung anerkannt gut randomisierte lange Schlüssel. Um die Erfindung bereits jetzt einsetzen zu können, wurde daher ein entsprechendes Verfahren zur randomisierten Erzeugung beliebig langer Schlüssel entwickelt. Das Verfahren ist im Ergebnis hardware-gestützt, da es bestimmte rechnerbasiert gewonnene „Vorratswerte“ voraussetzt. Das Verfahren wird in Fig. 3 gezeigt.

Die als Schlüssel-Chiffre-Paar kryptonisierten Ausgangsdaten werden dann über die Integrationseinheit so für die Zwischenspeicher aufbereitet, dass sie dort auslesbar und/oder abrufbar hinterlegt werden können, wobei immer eine Paar-Komponente zentral per DFÜ hinterlegt wird und die andere Paar-Komponente mobil.

Nach Hinterlegung der Daten wird/werden der/die mobilen Speicher physisch zum Zielsystem transportiert. Dort werden sie angeschlossen und von der Integrationseinheit ausgewertet, so dass schließlich die zentral hinterlegte Komponente abgerufen werden kann. Im Ergebnis stehen dann dem Zielsystem die kryptonisierten Ausgangsdaten zur Verfügung, so dass schließlich via Kryptoeinheit die Ausgangsdaten (S) rekonstruiert werden können.

Diese Vorgänge zur sicheren Datenübertragung sind in der Fig. 1 und Fig. 2 dargestellt.

Am Ausgangspunkt des technischen Ablaufs steht also die Kryptonisierung beliebiger Ausgangsdaten S in Form eines OTP- Schlüssel-Chiffre-Paares. Der Transport der Paar-

Komponenten erfolgt dann über mobile und zentrale Zwischenspeicher zum jeweiligen Zielsystem.

Diese Beschreibung des technischen Datenflusses trägt einer Besonderheit der OTP-Kryptographie Rechnung: Denn offenbar gilt $V+E=E+V$, d.h. **Schlüssel und Chiffre sind nicht unterscheidbar**, denn Schlüssel und Chiffre sind gleich lang, tragen keine Information und sind austauschbar.

Ein Extrembeispiel soll klarmachen, dass eine technische Datenfluss-Beschreibung für OTP-Schlüssel-Chiffre-Paare in „klassischen“ Begriffen irreführend wäre:

Angenommen, die Kryptoeinheit des Quellsystems liefert das Paar in der Weise, dass sie die Komponenten zufallsgesteuert vertauscht. Dann wüsste nicht einmal das Quellsystem, auf welchem Wege der Schlüssel und auf welchem die Chiffre transportiert wird.

Aber selbst wenn im Quellsystem dokumentiert würde, was Schlüssel und Chiffre war, wäre diese **Identifizierung eine Illusion**: Für ein gegebenes Schlüssel-Chiffre-Paar wäre diese Behauptung ja (mathematisch gesichert) nicht überprüfbar und damit technisch irrelevant.

In der Umkehrung gilt: Jede technische Datenfluss –Beschreibung, die als „Schlüssel“ bzw. „Chiffre“ bestimmte Datenelemente voraussetzt, erfasst OTP- Schlüssel- Chiffre –Paare nicht.

Der „**Identitätsverlust**“ von Schlüssel und Chiffre scheint auf den ersten Blick „paradox“. In der Tat liegt aber **kein Informationsverlust** vor, sondern vielmehr ein **Charakteristikum des vorgestellten sicheren Verfahrens**.

Gestützt auf den Hauptsatz der Informationstheorie (Shannon) könnte man sogar zeigen, dass jedes sichere Verfahren dieses Charakteristikum aufweisen muss.

Vorteilhaft ist also, dass ein zentraler Zwischenspeicher vorgesehen ist, der dadurch gekennzeichnet ist, dass ein Trägersystem dort verfahrensgemäß per DFÜ OTP-Schlüssel bzw. OTP-Chiffren abrufbar hinterlegt. Wie ausgeführt wäre es aber irreführend, von einem „Schlüssel-Pool“ oder einem „Chiffren-Pool“ zu sprechen. Es handelt sich vielmehr um einen **OTP-Datenpool**, d.h. einen Datenpool für Komponenten von OPT-Schlüssel-Chiffre-Paaren.

Ferner ist es vorteilhaft, mindestens ein Kommunikationsmodul vorzusehen, das mit dem OTP-Datenpool kommuniziert.

Vorteilhaft ist es ferner (und bei entsprechender Datenmenge zwingend), für die mobil gespeicherten Daten mindestens einen mobilen Massendatenspeicher (z.B. USB-Stick) vorzusehen.

Das erfindungsgemäße Verfahren bietet folgende Vorteile:

1. Mobile Massenspeicher sind inzwischen preiswert und über USB und ähnliche Schnittstellen problemlos einsetzbar, d.h. ohne zusätzliche Umrüstungskosten.
2. Wird der zentrale Massenspeicher nur als OTP-Datenpool genutzt, so sind bei geeigneter Referenzbildung (siehe Vorschläge unten) keinerlei Rückschlüsse auf die Ausgangsdaten S möglich.
3. Ferner könnten durch den Einsatz von Chipkarten in Kombination mit mobilen Massenspeichern (z.B. USB-Sticks) die mobilen Daten so verteilt werden, dass die mobil hinterlegte Paar-Komponente K_2 auf einem mobilen Massenspeicher („Faustpfand“) hinterlegt wird und die Referenz für die zentral hinterlegte Paar-Komponente K_1 auf der Chipkarte. Der mobile Massenspeicher könnte dann ohne Sicherheitsrisiken verloren gehen und die Chipkarte ohne Sicherheitsrisiko wie üblich

genutzt, d.h. ständig mitgeführt werden. Außerdem kann K_2 zusätzlich mit einem Kartenschlüssel verschlüsselt werden (siehe Beispiel unten)

Erwähnt sei auch, dass die Erfindung für die **biometrische Identifizierung von Personen** genutzt werden kann in einer Weise, die den Missbrauch der Identifizierungsdaten vermeidet. In diesem Szenario sind dann S biometrische Informationen, die das Quellsystem über ein entsprechendes Lesegerät erhält (z.B. einen Fingerprint- Scanner). Das Quellsystem hält S nur temporär, d.h. löscht S nach der Hinterlegung der kryptonisierten Daten wieder. Das gleiche gilt für das Zielsystem: Es rekonstruiert S über die kryptonisierten Daten, vergleicht S mit Eingangsdaten und löscht danach sowohl S wie die Vergleichsdaten. Im Ergebnis sind damit die kritischen Daten immer nur lokal und temporär verfügbar und dennoch ermöglicht das System die zweifelsfreie biometrische Identifizierung.

Diese unübersehbaren Vorteile gegenüber denen aus dem Stand der Technik bekannten Verfahren und Systemen machen die vorliegende Erfindung darüber hinaus auch zukunftsicher.

Im nun Folgenden wird die Erfindung anhand von Zeichnungen im Einzelnen näher beschrieben. Es zeigt

Fig. 1: eine Blockdarstellung der Zusammenhänge der verschiedenen Module in dem erfindungsgemäßen Daten-Trägersystems (1);

Fig. 2: ein Blockschaltbild des Daten-Trägersystems (1) mit den dazugehörigen verschiedenen technischen Einrichtungen.

Fig. 3: ein Verfahren zur Rechner-gestützten randomisierten Erzeugung langer Schlüssel.

Die Fig. 1 zeigt ein Blockdarstellung der Zusammenhänge der verschiedenen Module in dem erfindungsgemäßen Daten-Übertragungs-System 1. Ausgehend von (IT-) Systemen, welche personenbezogene bzw. Eigentümer-bezogene Daten speichern, wird ein Integrationssystem vorgestellt, das den sicheren, insb. fälschungssicheren, Austausch dieser Daten zwischen den Ausgangssystemen so ermöglicht, dass die betroffenen Personen die letztendliche Kontrolle über den Datenaustausch behalten.

Die Eigentümer der Ausgangssysteme werden nachfolgend als **Träger** bezeichnet, ihre IT-Systeme als **Trägersysteme**.

Die Eigentümer der personenbezogenen Daten werden als **Betroffene** bezeichnet.

Ein möglicher Anwendungsbereich für diese Lösung ist das Gesundheitswesen (Stichwort: Elektronische Gesundheitskarte/ egk). Dort sind die medizinischen Leistungserbringer die Träger, Trägersysteme sind IT- Systeme dieser Leistungserbringer (bei einem Arzt z.B. die Arzt-Software mit zugehöriger Hardware). Betroffene sind Patienten bzw. Versicherte.

Wir setzen nachfolgend voraus, dass jeder Träger bzw. jedes Trägersystem systemweit geeignet identifiziert wird. Sinnvollerweise wird die Identifizierung über die Zeit persistent gehalten, z.B. durch generell fortlaufende Nummerierung, so dass eine Nummer über die Zeit nur einmal vergeben wird.

Die Identifizierung der Trägersysteme bzw. der Träger ist konstitutiv für die Abgrenzung des Gesamtsystems (**Ensemble-Identifizierung**) und daher wichtig. Eine Identifizierung/ Registrierung der mobilen Speicher bzw. der Betroffenen wird nicht vorausgesetzt. Es ist ein besonderer Vorteil der Erfindung, dass diese völlig anonym bleiben können.

Das neue Gesamtsystem entsteht durch Erweiterung eines Ensembles von Trägersystemen um zentrale Speicher und ortsungebundene (portable) personenbezogene Speicher für Betroffene, sowie einer Logik, welche alle Speicher miteinander vernetzt.

Betroffene, die am System teilnehmen wollen, werden dazu mit folgenden Speicherelementen ausgestattet, wobei die zugehörigen Lesegeräten / Schnittstellen ebenfalls genannt werden:

- Eine oder mehrere beschreibbare Chipkarten mit üblichen Kartenschreiblesegeräten, kurz Kartengeräte genannt, als Schnittstelle.
- Einen oder mehrere portable Massenspeicher mit USB- Schnittstelle, nachfolgend USB-Sticks genannt, wobei die USB-Technik hier beispielhaft für eine Datenspeicherzugriffstechnik steht. Andere portable Massenspeicher-Lösungen mit hinreichender Verbreitung sind auch denkbar.

Bei entsprechenden Datenmenge ist der Einsatz mindestens eines mobilen Massenspeichers pro Betroffenen konstitutiv, d.h. eine Minimalvoraussetzung. Der Einsatz einer beschreibbaren Karte wird empfohlen.

Ein weiteres zentrales Speichersystem komplettiert die Infrastruktur:

- Ein Zentralspeicher (nachfolgend **OTP-Register** genannt) mit Internet-Schnittstelle bzw. analoger Schnittstellen für den Datenfernzugriff, auf den über entsprechende Protokolle wie z.B. https in autorisierter Form zugegriffen wird; es seien auch Lösungen erfasst, die mehrere aus Sicht der Trägersysteme zentrale OTP-Register vorsehen; für diesen Fall nehmen wir an, dass diese über eindeutige Registernummern identifizierbar sind.

Diese neuen technischen Elemente und die Trägersysteme werden über eine Logik integriert, die z.B. über Software-Komponenten realisiert werden kann und nachfolgend als **Module** bezeichnet werden.

Die Logik wird funktional beschrieben über entsprechende Funktionsgruppen. Die Gruppenbildung ist nahe liegend, aber nicht zwingend.

In diesem Sinne wird die Steuereinheit, welche auf einem Trägersystem die Logik repräsentiert, gestützt auf weitere Funktionseinheiten, nachfolgend als **Integrationsmodul** bezeichnet.

Das Integrationsmodul stützt sich auf folgende weitere Funktionsgruppen:

- Das **Kommunikationsmodul** kommuniziert mit dem zentralen OTP-Register
- Das **Kryptomodul** enthält die Verschlüsselungstechnik inkl. Zufallsgenerator
- Das **Verpackungsmodul** serialisiert / deserialisiert Daten
- Das **Kartenmodul** erzeugt/ interpretiert die Kartendaten, d.h. bereitet insbesondere die via Kartengerät ausgelesenen Daten so auf, dass nachgelagerte Systeme sie verarbeiten/ darstellen können.
- Das **USB-Modul** erzeugt/ interpretiert Daten auf den USB-Sticks.

Kommunikationsmodul, Kryptomodul, USB-Modul sind auf allen Trägersystemen funktionsgleich. Somit sind sie portabel, d.h. ortsungebunden in das System implementierbar (z.B. als SW-Module).

Die Verpackungsmodule sind in ihren Eingaben naturgemäß Trägersystem-spezifisch, z.B. spezifisch für eine Arzt-Software.

Die Kartenmodule sind naturgemäß im Kern in ihren Schnittstellen einheitlich (Kartenmodulkern). Für die Integration eines solchen Kerns in einem Trägersystem sind aber ggf. weitere Trägersystem- spezifische Schnittstellen nützlich.

Die nachfolgend beschriebene beispielhafte Realisierung eines erfindungsgemäßen Systems basiert auf Chip-Karten und USB- Stricks. Um einen realitätsnahen Vergleich mit der egk- Lösung zu ermöglichen wird mit einer verfeinerten Sicht auf die Daten der Karte begonnen. Die Karte enthält

- einen Schlüssel für das AES-Verfahren (Advanced-Encryption-Standard), vom Kryptomodul im Rahmen der dezentralen Initialisierung randomisiert erzeugt, nachfolgend **Kartenschlüssel** genannt; AES steht hier beispielhaft für ein symmetrisches Verschlüsselungsverfahren.
- beliebige Grunddaten über Betroffene, z.B. Angaben zur Personenidentifizierung, Blutgruppe, Versicherungsnummern, werden nachfolgend als **Basisdaten** bezeichnet;
- Metadaten über den/ die USB-Sticks, die nachfolgend als **Stickregister** bezeichnet werden.

Basisdaten sind nicht konstitutiv, d.h. es ist im Extremfall auch vorstellbar, dass keine Grunddaten über Betroffene benutzt werden.

Metadaten sind nicht konstitutiv, aber sehr sinnvoll. Z.B. wäre die Anzahl und Größe der USB-Objekte eine sinnvolle Metainformation oder verfeinert Kontextinformationen für die jeweilige Serialisierung, inkl. Kontrollwerte.

Der AES- Schlüssel ist nicht konstitutiv, d.h. kann entfallen. Er wird aber empfohlen.

Wichtig: Eine Karte und ihre zugehörigen USB-Sticks werden vom jeweiligen Betroffenen verwaltet, der bzw. die Zentralspeicher von einer bzw. mehreren zentralen Instanzen, Trägersysteme wie bisher von ihrem Träger.

Das entstehende Gesamtsystem leistet, dass die Daten sicher und insb. fälschungssicher und unter Kontrolle des Betroffenen von einem Trägersystem A zu einem Trägersystem B übertragen werden können, z.B. von einer Patientenakte von einem Facharzt zum Hausarzt. Für ein gegebenes serialisierbares Datenobjekt D bereitet das Trägerquellsystem, d.h. das Trägersystem, auf dem die Daten erstmalig entstehen, den Datenaustausch wie folgt vor (**Virtualisierung des Datenobjekts**):

1. D wird vom Verpackungsmodul entsprechend serialisiert, d.h. in eine entsprechende Bytefolge S umgewandelt (nachfolgend **Speicherobjekt** genannt).
2. Das Integrationsmodul nummeriert das Speicherobjekt S mit einer fortlaufenden Nummer, bzw. alternativ mit einer eindeutigen Zufallszahl und wird nachfolgend lokale Objektnummer genannt, so dass das Tupel (lokale Objektnummer, Trägernummer, Trägersystemsnummer, OTP-Registernummer) eine **systemweit eindeutige Referenz** für das Speicherobjekt S ist.
3. Das Speicherobjekt S wird vom OTP-Kryptomodul als OTP-Schlüssel-Chiffre-Paar (K_1 , K_2) dargestellt, wobei K_1 = Schlüssel bzw. Chiffre und K_2 = Chiffre bzw. Schlüssel.
4. In einer zweiten **Verschlüsselung** wird anschließend K_2 zusätzlich vom Kryptomodul mit dem Kartenschlüssel verschlüsselt (Ergebnis: K_2'). Diese Verschlüsselung ist nicht konstitutiv, d.h. wird lediglich empfohlen.

5. Die Paar-Komponenten K_1 wird nun zusammen mit der S-Referenz via Kommunikationsmodul auf das zentrale OTP-Register gem. OTP-Registernummer übertragen.
6. Die (zusätzlich verschlüsselte) Paar-Komponente K_2' wird als Binärdatei auf den USB-Stick kopiert (**Personalisierung 1**).
7. Das Stickregister wird aktualisiert ; ggf. ebenso assoziierte Verwaltungsinformationen auf der Karte oder dem USB-Stick (**Personalisierung 2**)
8. Ergebnis: Das Datenobjekt D ist somit virtualisiert, d.h. die kryptonisierten Daten (K_1, K_2) sind verfahrensgemäß außerhalb des Quellsystems hinterlegt.

Dargestellt wurde die Virtualisierung der Daten D auf Basis einer initialisierten Karte mit Kartenschlüssel ,etc. Ist die Karte nicht initialisiert, ist der Ablauf lediglich um einen weiteren Initialisierungsschritt zu ergänzen, der ebenfalls im Trägerquellsystem, gestützt auf das Kartenmodul, abläuft (Erzeugung des Kartenschlüssels, Laden der Basisdaten, usw.).

Die Lösung benötigt also keine zentrale Initialisierung.

Der Betroffene könnte – je nach Wunsch – einen oder mehrere USB-Sticks verwenden, sofern das Stickregister hinreichend flexibel konzipiert ist.

Im Trägersystem eines anderen Trägers, nachfolgend Trägerzielsystem genannt, wird umgekehrt vorgegangen, um das Speicherobjekt S zu übertragen:

1. (Datenanschluss) Der Betroffene führt in der Zielumgebung seine Karte und den zugehörigen (bzw. einen zugehörigen) USB-Stick in entsprechende Lesegeräte ein.

2. (Auswahlvorgang) Der Träger bzw. ein stellvertretender Bearbeiter bestimmt via Kartenmodul über das Stickregister ein virtuelles Speicherobjekt S. Dessen S – Referenz und das verschlüsselte K_2' wird dann vom USB-Stick auf das Trägersystem kopiert (inkl. Kartenschlüssel).

Empfehlung: Sinnvollerweise wird dieser Auswahlvorgang über entsprechende Systeme unterstützt; über Metainfo auf der Karte bzw. dem USB-Stick könnte dann sichergestellt werden, dass die Auswahl entsprechend eingeschränkt wird. Bei med. Leistungserbringern z.B. über das Fachgebiet.

3. Über das Kommunikationsmodul wird autorisiert via Referenz auf den Zentralspeicher zugegriffen und die komplementäre Paarkomponente K_1 abgerufen (zentraler Komponentenzugriff).
4. Via Kryptomodul wird zunächst das verschlüsselte K_2' mit dem Kartenschlüssel entschlüsselt und anschließend aus den kryptonisierten Ausgangsdaten K_1, K_2 via Kryptomodul die Ausgangsdaten S rekonstruiert (Entschlüsselung).
5. Im Ergebnis liegt S auf dem Zielsystem.

Die Fig. 2 zeigt ein Ausführungsbeispiel der vorliegenden Erfindung als empfohlenes Ausführungsbeispiel:

- Der USB-Eingang und der Karten-Eingang sind im Trägersystem gesicherte und getrennte Kanäle, die in einen (gesicherten) Kernbereich führen.
- Der Weg zu diesem Kernbereich wird mit einer zusätzlichen, sitzungsbezogenen Verschlüsselung abgewickelt

- Jeder Zugriff wird im Zentralspeicher vermerkt, sodass bei einer Revision eines Trägersystems festgestellt werden könnte, ob nur zulässige Informationen heruntergeladen wurden.

Die Erfindung gestattet also den sicheren und authentischen (fälschungssicheren) Austausch personenbezogener Daten **ohne dass die Betroffenen selbst im System in irgendeiner Weise identifiziert sein müssen**. Die kryptonisierten Daten K_1 bzw. K_2 sind ferner **keine Daten im klassischen Sinne: sie tragen keine Information**. Die Lösung ist demnach zukunftssicher, muss also z.B. auch den Quantencomputer nicht fürchten.

Der Vollständigkeit halber sei auch die Bit-Addition kurz ausgeführt und die für die Chiffrierung relevanten Eigenschaften nachgewiesen. Es wird eine „programmernahe Notation“ verwendet. Anstelle des +Zeichens wird das ^Zeichen für den entsprechenden Bit-Operator verwendet, der in vielen Programmiersprachen zur Verfügung steht.

Definition: Sei $!0=1$, $!1=0$ (**Negation**)

Es gilt dann offenbar $!!x=x$

Definition: Sei $0^0=1^1=0$ und $0^1=1^0=1$ (**Bit-Addition bzw. XOR-Verknüpfung**)

Dann gilt stets

- $x^x=0$ (klar)
- $x^{!x}=1$ (klar)
- $x^0=x$ (denn $1^0=1$, $0^0=0$)
- $x^1=!x$ (denn $1^1=0$, $0^1=1$)

Ferner gilt für zwei Bitvariable x, y stets:

- $x^y=y^x$ (klar)

Wir betrachten nun beliebige Bitvariable x, y, z und zeigen:

- $x^{(y^z)} = (x^y)^z$

Annahme: $y = z$.

Die rechte Seite liefert dann

- $x^{(y^z)} = x^0 = x$

Für die linke Seite sind zwei Fälle möglich:

- $(x^x)^x = 0^x = x$
- $(x^{!x})^{!x} = 1^{!x} = !x$

Annahme: $y \neq z$.

Für die linke Seite sind dann folgende Fälle möglich:

- $x^{(x^{!x})} = x^1 = !x$
- $x^{(!x^x)} = x^1 = !x$

Für die rechte Seite sind dann folgende Fälle möglich:

- $(x^{!x})^x = 1^x = !x$
- $(x^x)^{!x} = 0^{!x} = !x$

Folglich gilt die Gleichung in allen Fällen.

Sei also S wie bisher eine Bit-Liste, E ein gleich langer Einmalschlüssel, und die Chiffre V sei definiert als $V = S^E$, wobei komponentenweise addiert wird. Ist dann O eine gleich lange Bitliste mit lauter Nullen, dann gilt $V^E = (S^E)^E = S^{(E^E)} = S^O = S$.

Abschließend sei das Randomisierungsverfahren nach Fig. 3 in seinen wesentlichen Aspekten beschrieben:

Eine lange Bitliste wird mit einem Standard-Zufallsgenerator „stückweise“ erzeugt, wobei der Zufallsgenerator nach jedem Schritt mit **sicher verschlüsselten Vorratswerten** reinitialisiert wird. Auch die Bestimmung der Stücklänge und die Auswahl der Vorratswerte erfolgt randomisiert. Sind die Stücke „kurz genug“ und ist der Wertevorrat „groß genug“ und „unvorhersehbar genug“, dann wird **im Ergebnis eine Folge unabhängiger Zufallsexperimente simuliert**: Sicher verschlüsselte Werte sind offenbar ideale Reinitialisierungswerte, so dass sich die Unabhängigkeit gewissermaßen „Schritt für Schritt

vererbt“. Das Verfahren liefert also unter den genannten Voraussetzungen randomisierte Bitlisten von hoher Qualität. Geeignete Vorratswerte können Rechner-gestützt gewonnen werden (siehe Fig. 3). Mit einem solchen Wertevorrat wäre auch die Menge möglicher Ergebnisse so groß, dass sie extern nicht mehr simulierbar wäre.

PATENTANSPRÜCHE

1. Verfahren zum kontrollierten Datenaustausch zwischen Datenträgersystemen mittels mobiler (ortsungebundener) Schreib-Lese-Speicher, dadurch gekennzeichnet, dass ein erstes Trägersystem (**Quellsystem**) beliebige serielle Ausgangsdaten (S) via Kryptomodul als **ONE-TIME-PAD-Schlüssel-Chiffre-Paar** darstellt, wobei die Paar-Komponenten nicht als Schlüssel bzw. Chiffre identifiziert sein müssen und dann extern zwischenspeichert, wobei immer eine Paar-Komponente (K_1) mit systemweit geeigneter Referenz via DFÜ auf einem zentralen Zwischenspeicher (**OTP-Datenpool**) abrufbar hinterlegt wird und die andere Paar-Komponente (K_2) und die K_1-Referenz auf mobilen Speichern, so dass ein zweites Trägersystem (**Zielsystem**) durch Anschluss und Auswertung der mobilen Speicher und durch Abruf der zentral hinterlegten Komponente zunächst das OTP-Schlüssel-Chiffre-Paar und schließlich durch Entschlüsselung die Ausgangsdaten selbst gewinnen kann.
2. Trägersystem (1) zum kontrollierten Datenaustausch zwischen Datenträgersystemen, dadurch gekennzeichnet, dass es mit einem OTP-Kryptomodul ausgestattet ist und (a) als Quellsystem beliebige serielle Ausgangsdaten (S) in Form von OTP-Schlüssel-Chiffre-Paaren kryptonisiert bzw. (b) als Zielsystem aus OTP-Schlüssel-Chiffre-Paaren Ausgangsdaten rekonstruiert.
3. Zentraler Zwischenspeicher nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass ein Trägersystem dort verfahrensgemäß per DFÜ OTP-Schlüssel bzw. OTP-Chiffren abrufbar hinterlegt bzw. abruft.

4. Ortsungebundener (mobiler) Zwischenspeicher nach einem der vorangegangenen Ansprüche, d a d u r c h g e k e n n z e i c h n e t, dass ein Trägersystem dort verfahrensgemäß OTP-Schlüssel bzw. OTP-Chiffren abspeichert bzw. ausliest.

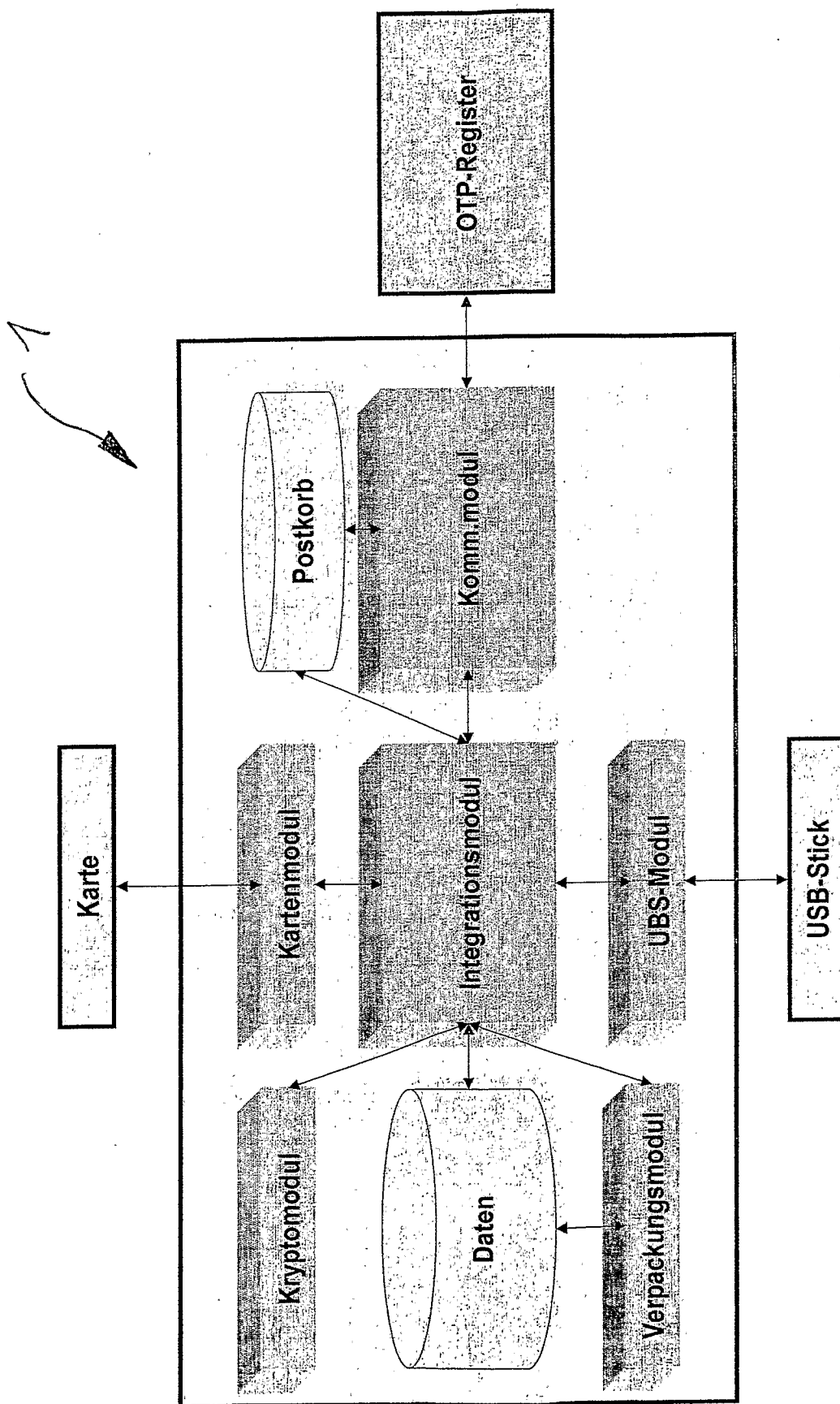


Fig. 1

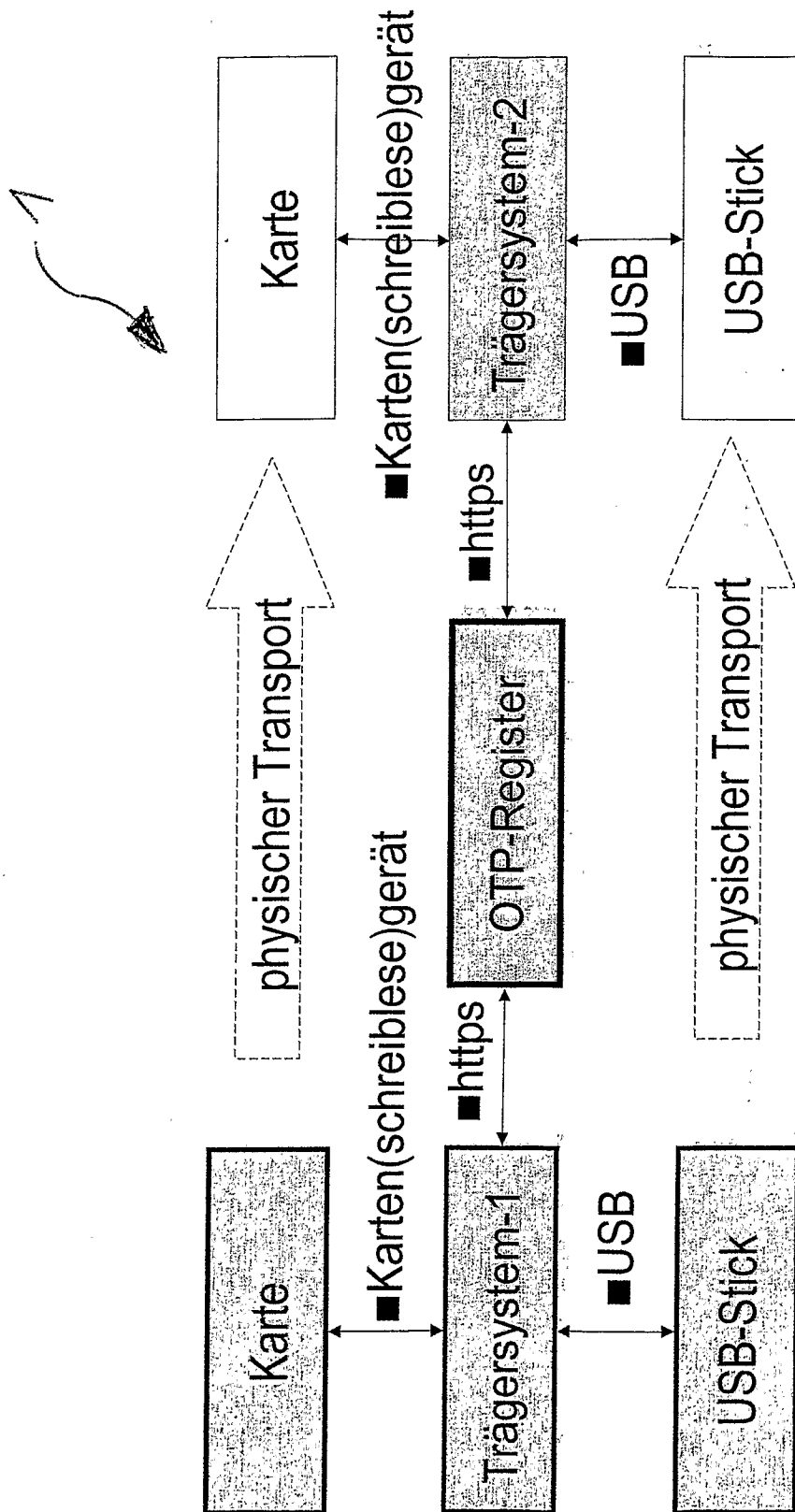


Fig. 3

Randomisierte Erzeugung langer Bitliste

Robert Niggel

1 Zusammenfassung

Dieses Dokument beschreibt ein Verfahren zur Erzeugung beliebig langer randomisierter Bitlisten auf Basis von Zufallsgeneratoren für Bitlisten mit begrenzter Bitzahl.

Die Strategie besteht darin, einen entsprechenden Vektor $B = (B_1, \dots, B_n)$ von **hinreichend kurzen Teillisten B_i zu erstellen, die durch unabhängige Reinitialisierung gewonnen werden**, und zwar so, dass der Reinitialisierungsprozess über eine variable Anzahl von Parametern (Vorratswerte genannt) gesteuert wird.

Es wird gezeigt, dass unter bestimmten Voraussetzungen – insb. hinsichtlich der Zahl und der Gewinnung der Vorratswerte – die Gesamtliste hinreichend gut randomisiert ist.

2 Grundbegriffe, Notationen

Definition 1. Ein (elementarer) Zufallsgenerator Z wird nachfolgend beschrieben als Tupel (f, g, m) mit

- f ist eine Initialisierungsfunktion
- g ist eine Produktionsfunktion die m Bits liefert

Für einen Zufallsgenerator $Z = (f, g, m)$ sei

- $Init(Z) := f$ (sprich: **Initialisierungsfunktion von Z**)
- $Prod(Z) := g$ (sprich: **Produktionsfunktion von Z**)
- $Bitzahl(Z) := m$ (sprich: **Bitzahl von Z**)
- $Inits(Z) := \text{Anzahl der Parameter von } Init(Z)$

Definition 2. Ist Z ein Zufallsgenerator, so sei $Byte(Z)$ folgende Funktion:

- Eingabe: Eine Zahl h
- Ausgabe: h Byte, die durch wiederholten Aufruf von $Prod(Z)$ erzeugt werden, d.h. $Prod(Z)$ wird x -mal, $Bitzahl(Z)*x \geq h*8$, hintereinander aufgerufen, die erzeugten Bits verkettet und dann (die ersten) h Byte ausgegeben.

Definition 3. Für eine Liste $a = (a_1, \dots, a_m)$ sei $|a| := m$ die Listenlänge. Sind $a = (a_1, \dots, a_m), b = (b_1, \dots, b_n)$ Listen, so sei $ab := (a_1, \dots, a_m, b_1, \dots, b_n)$ diejenige Liste, die sich durch Verkettung von a mit b ergibt. Ist $X = (X_1, \dots, X_M)$ eine Liste von Listen, so sei $Liste(X)$ die Verkettung aller Elemente:

- $Liste(X) := X_1$ für $M = 1$
- $Liste(X) := Liste((X_1, \dots, X_{M-1}))X_M$ für $M > 1$

Definition 4. Für ein Datenobjekt x sei nachfolgend $dim(x)$ der Speicherbedarf von x in Byte.

3 Aufgabenstellung

Mit einem Zufallsgenerator Z ist eine Byteliste der (Mindest)Länge M zu erzeugen.

4 Lösung

4.1 Übersicht

Es wird via Z schrittweise ein Vektor $B = (B_1, \dots, B_n), n > 0$, von Bytelisten B_i erzeugt, die insgesamt eine ausreichend lange Gesamtliste ergeben, d.h.

- $|Liste((B_1, \dots, B_{n-1}))| < M \leq |Liste(B)|$

wobei vor jedem Schritt die Initialisierungsfunktion von Z aufgerufen wird:

- Beim ersten Schritt ($i=1$) werden klassische Initialisierungswerte verwendet (z.B. aktuelle Zeitstempel).
- In jedem Schritt i werden die Reinitialisierungsparameter für den nächsten Schritt ($i+1$) bestimmt dergestalt, dass aus einem Wertevorrat $V = (V_1, \dots, V_H)$ eine entsprechende Auswahl von Werten V_j randomisiert erfolgt und diese Parameter dann One-Time-Pad-verschlüsselt werden, wobei die Schlüssel jeweils via $Byte(Z)$ neu erzeugt werden.
- Ferner wird die Länge von B_i im Schritt i randomisiert innerhalb einstellbarer Grenzen – eine Mindestlänge (L) bzw. Maximallänge (L') – bestimmt.

4.2 Algorithmus

```

1: procedure RANDOMIZE( $Z, B, M, L, L', V$ )
2:   // Berechnet  $B$  mit  $|Liste(B)| \geq M$  via  $Z$  unter folgenden Voraussetzungen:
3:   //  $M > 0$ 
4:   //  $0 < L \leq L'$ 
5:   //  $|V| > 0$ 
6:   // Anweisungen  $x := Byte(Z)(dim(x))$  setzen nachfolgend einen  $x$ -Typ voraus,
7:   // der mit beliebigen Bits belegt werden kann (z.B. unsigned int)
8:    $B$  initialisieren // nun gilt:  $B = (B_1, \dots, B_n), n = 0$  also  $|B| = 0, |Liste(B)| = 0$ 
9:   Hilfsvektor  $W, |W| = Inits(Z)$ , mit Standardinitialisierungswerten belegen
10:  for  $|Liste(B)| < M$  do // Schritt:  $B = (B_1, \dots, B_n)$  um ein Element erweitern
11:     $Init(Z)$  mit  $W$  aufrufen // Initialisierung von  $Z$ 
12:    for  $k = 1..Inits(Z)$  do //  $W_k$  für die Reinitialisierung belegen
13:       $r := Byte(Z)(dim(r))$  // Zufallszahl  $r$  bestimmen
14:       $h := mod(r, |V|)$  // Vorrantsindex  $h$  bestimmen
15:       $S := Byte(Z)(dim(V_h))$  // Schlüssel  $S$  bestimmen
16:       $W_k := XOR(V_h, S)$  //  $V_h$  OTP-verschlüsseln
17:       $r := Byte(z)(dim(r))$  // weitere Zufallszahl  $r$  bestimmen
18:       $d := max(L, mod(r, L'))$  // Länge  $d$  des neuen Elementes festlegen
19:       $b := Byte(Z)(d)$  // Byteliste  $b$  der Länge  $d$  berechnen
20:       $B := (B_1, \dots, B_n, b)$  //  $B$  um  $b$  erweitern
21:  // nun gilt:  $|B| > 0, |Liste(B)| \geq M$ 

```

4.3 Bewertung

Annahmen:

1. Die Teillisten sind kurz genug (zu steuern über L, L').
2. Die Anzahl der Vorratswerte ist groß genug (z.B. $|V| = 10000$).
3. Die Vorratswerte entstehen auf dem Rechner systemtechnisch bedingt und ohne Zusammenhang mit irgendwelchen Fachdaten so, dass sie extern nicht vorhersehbar sind.

Unter diesen Voraussetzung ist dann B hinreichend gut randomisiert:

- Die erste Teilliste wird durch klassische Initialisierung gewonnen, ist also hinreichend gut randomisiert.
- In jedem Folgeschritt wird der Zufallsgenerator mit Werten reinitialisiert, die hinreichend gut randomisiert sind, also ist jede Teilliste eines Folgeschrittes hinreichend gut und unabhängig randomisiert, denn:

Werden im Schritt i die ausgewählten Vorratswerte sicher verschlüsselt, dann werden auch im Schritt $i + 1$ die ausgewählten Vorratswerte sicher und unabhängig verschlüsselt.

- Es ist praktisch nicht möglich, die Menge möglicher Ergebnisse über die Menge möglicher Eingabewerte zu simulieren.

5 Vorratsbestimmung

Implementiert:

- Sammeln von Adressen von allokierten Objekten in einem Vektor.
- Ist die maximale Vektorlänge erreicht, so werden die ältesten Einträge überschrieben, d.h. es wird eine Schreibposition geführt, die nach Erreichen der Maximallänge wieder auf 0 gesetzt wird.