(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
1. August 2013 (01.08.2013)

WIPO | PCT

(10) Internationale Veröffentlichungsnummer
WO 2013/110103 A2

(51) Internationale Patentklassifikation:
H04L 9/08 (2006.01)

(21) Internationales Aktenzeichen: PCT/AT2013/000010

(22) Internationales Anmeldedatum:
22. Januar 2013 (22.01.2013)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
A 106/2012 26. Januar 2012 (26.01.2012) AT

(72) Erfinder; und
(71) Anmelder : **CORDES, René-Michael** [AT/AT]; Raiffeisengasse 3, A-2323 Mannswörth (AT). **SCHOBESBERGER, Ernesto** [AT/AT]; Prinz Eugen Straße 52/9, A-1040 Wien (AT).

(74) Anwalt: **KESCHMANN, Marc**; Haffner und Keschmann Patentanwälte KG, Schottengasse 3a, A-1014 Wien (AT).

(81) Bestimmungsstaaten *(soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, RU, TJ, TM), europäisches (AL, 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, RS, SE, SI, SK, SM, 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: METHOD FOR WRITING AND READING DATA

(54) Bezeichnung : VERFAHREN ZUM SCHREIBEN UND LESEN VON DATEN

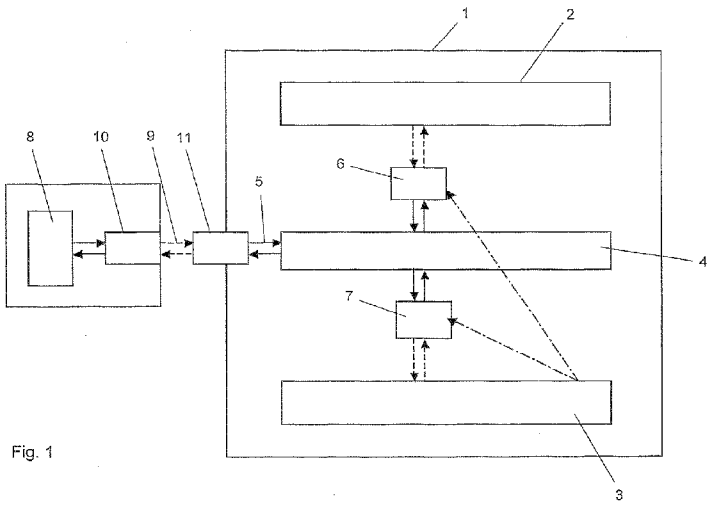


Fig. 1

(57) Abstract: The invention relates to a method for writing and reading data into and out of an indexed data base (1) which comprises a data structure (2) and an associated index structure (3). In said method, a processing unit (4) receives data to be written in plain text and, by means of a write access, writes the data into the data structure (2) and updates index data in the index structure (3). The processing unit (4) determines readout data or the memory location thereof by means of access to the index data (3) and reads the readout data out of the data structure (2) by means of a read access and provides said data in plain text. The data in the data structure (2) and the index data in the index structure (3) are stored encrypted, wherein the write/read access of the processing unit (4) to the index structure (3) and to the data structure (2) is effected via at least one encryption and decryption unit (6, 7), by which the data is encrypted and decrypted by means of stream ciphering.

(57) Zusammenfassung:

[Fortsetzung auf der nächsten Seite]

WO 2013/110103 A2

Bei einem Verfahren zum Schreiben und Lesen von Daten in einen bzw. aus einem indizierten Datenbestand (1), der eine Datenstruktur (2) und eine zugehörige Indexstruktur (3) umfasst, empfängt eine Verarbeitungseinheit (4) zu schreibende Daten im Klartext und schreibt die Daten mittels eines Schreibzugriffs in die Datenstruktur (2) und aktualisiert Indexdaten in der Indexstruktur (3). Die Verarbeitungseinheit (4) ermittelt auszulesende Daten oder deren Speicherort mittels eines Zugriffs auf die Indexdaten (3) und liest die auszulesenden Daten mittels eines Lesezugriffs aus der Daten- struktur (2) aus und stellt diese im Klartext zur Verfügung. Es. werden die Daten in der Datenstruktur (2) und die Indexdaten in der Indexstruktur (3} verschlüsselt gespeichert, wobei der Schreib-/Lesezugriff der Verarbeitungseinheit (4) auf die Indexstruktur (3) und auf die Datenstruktur (2) über mindestens eine Ver- und Entschlüsselungseinheit (6, 7) erfolgt, mit der die Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselt werden.

Verfahren zum Schreiben und Lesen von Daten

Die Erfindung betrifft ein Verfahren zum Schreiben und Lesen von Daten in einen bzw. aus einem indizierten Datenbestand, der eine Datenstruktur und eine zugehörige Indexstruktur umfasst, wobei eine Verarbeitungseinheit zu schreibende Daten im Klartext empfängt und mittels eines Schreibzugriffs in die Datenstruktur schreibt und Indexdaten in der Indexstruktur aktualisiert und wobei die Verarbeitungseinheit auszulesende Daten oder deren Speicherort mittels eines Zugriffs auf die Indexdaten ermittelt und die auszulesenden Daten mittels eines Lesezugriffs aus der Datenstruktur ausliest und im Klartext zur Verfügung stellt.

Die Erfindung betrifft weiters eine Vorrichtung zum Schreiben und Lesen von Daten in einen bzw. aus einem indizierten Datenbestand, der eine Datenstruktur und eine zugehörige Indexstruktur umfasst, umfassend eine Verarbeitungseinheit, in der zu schreibende Daten im Klartext empfangen werden können und die einen Schreibzugriff auf die Datenstruktur aufweist, um die Daten in die Datenstruktur zu schreiben, und die mit der Indexstruktur zusammenwirkt, um Indexdaten in der Indexstruktur zu aktualisieren, und die einen Zugriff auf die Indexdaten aufweist, um auszulesende Daten oder deren Speicherort zu ermitteln, und die einen Lesezugriff auf die Datenstruktur aufweist, um die auszulesenden Daten aus der Datenstruktur auszulesen und im Klartext zur Verfügung zu stellen.

Gegenwärtig stellen indizierte Datenbestände, insbesondere indizierte Datenbanken die meist verbreitete Massenspeicherung von Daten dar. Aus hardwaretechnischer Sicht ist eine indizierte Datenbank ein Massenspeicher, welcher einen Indexspeicher zur Beschleunigung des Zugriffs angeschlossen hat.

Ein Datenbankindex ist eine von der Datenstruktur getrennte Indexstruktur in einer Datenbank, welche die Suche und das Sortieren nach bestimmten Feldern beschleunigt. Ein Index besteht aus einer Ansammlung von Zeigern (Verweisen), die eine Ordnungsrelation auf eine oder mehrere Spalten in einer Tabelle definieren. Wird bei einer Abfrage eine indizierte Spalte als Suchkriterium herangezogen, sucht eine Verarbeitungseinheit, d.i. in der Regel ein Datenbankmanagementsystem, die gewünschten Datensätze anhand dieser Zeiger. Ohne Index müsste die Spalte sequentiell durchsucht werden, was selbst mit schneller Hardware viel Zeit in Anspruch nimmt. Es gibt eine Vielzahl unterschiedlicher Indexstrukturen. In der Regel finden jedoch B^+-Bäume Anwendung.

Es ist wünschenswert, Datenbanken zu verschlüsseln, um den Zugriff auf den Inhalt der Datenbank vor unbefugten Zugriffen zu schützen. Dabei soll der schnelle Zugriff auf die verschlüsselten Daten aber erhalten bleiben, d.h. es soll vermieden werden, die gesamte Datenbank entschlüsseln zu müssen, bevor mittels einer Suchabfrage nur auf einen oder mehrere bestimmte Datensätze zugegriffen wird. Es wäre daher wünschenswert, dass jeweils nur die auszulesenden Datensätze entschlüsselt bzw. die zu schreibenden Datensätze verschlüsselt werden.

Ein sicherer Schutz der Datenbank ist nur dann gewährleistet, wenn nicht nur die in der Datenstruktur abgelegten Daten, sondern auch die in der Indexstruktur gespeicherten Indexdaten verschlüsselt vorliegen.

Wenn Daten verschlüsselt werden, ist eine funktionelle Anforderung, dass diese in einen eindeutigen Zusammenhang zu dem zur Verschlüsselung verwendeten Code gestellt werden können.

Beim Indizieren von verschlüsselten Daten treten folgende Probleme auf: Beim Verschlüsseln der Daten muss der Inhalt der Daten von der Verarbeitungseinheit erkannt werden können. Weiters muss berücksichtigt werden, ob bei der Verschlüsselung der Daten deren Größe auf dem Speichermedium verändert wird. Beim Entschlüsseln der Daten ist sicherzustellen, dass der zur Verschlüsselung verwendete Schlüssel wieder zur Verfügung steht.

Die Erfindung zielt daher darauf ab, ein Verfahren und eine Vorrichtung zu schaffen, mit dem bzw. der die Vertraulichkeit der Datenstruktur und der Indexstruktur eines Datenbestandes geschützt werden kann, ohne dass der Zugriff auf die Daten unter Verwendung des Index durch berechtigte Benutzer beeinträchtigt ist. Es soll die uneingeschränkte Funktionalität einer indizierten Datenbank aufrechterhalten werden.

Zur Lösung dieser Aufgabe ist gemäß einem ersten Aspekt der Erfindung vorgesehen, dass die Daten in der Datenstruktur und die Indexdaten in der Indexstruktur verschlüsselt gespeichert werden und dass der Schreib-/Lesezugriff der Verarbeitungseinheit auf die Indexstruktur und auf die Datenstruktur über mindestens eine Ver- und Entschlüsselungseinheit erfolgt, mit der die Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselt werden. Dadurch, dass die zu schreibenden und zu lesenden Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselt werden, ist sichergestellt, dass das Abbild der verschlüsselten Daten und der unverschlüsselten Daten auf dem Speichermedium exakt dieselben Ausmaße (Bitlänge) hat, so dass sie auch in verschlüsselter Form bitgenau aufgefunden werden und an den anfordernden User in Unkenntnis des Inhalts übermittelt werden können. Da jede einzelne Information exakt dieselben Ausmaße (Bitlänge) wie die unverschlüsselte auf-

weist, kann von einem in unverschlüsselter Form angefertigten Index auch auf die Position der verschlüsselten Daten exakt zugegriffen werden, so dass der Inhalt der verschlüsselten Daten von der Verarbeitungseinheit der Datenbank nicht erkannt werden muss und auf den Umstand der Verschlüsselung bei der Speicherplatzdurchsuchung keine Rücksicht genommen werden muss.

Als Stromverschlüsselung bezeichnet man einen kryptographischen Algorithmus, bei dem Zeichen des Klartextes mit den Zeichen eines Schlüsselstroms einzeln verknüpft werden. Im Fall der Stromverschlüsselung von digitalen Daten - es kommen nur die Zeichen 0 und 1 zum Einsatz - erfolgt die Verknüpfung des Klartextstroms mit dem Schlüsselstrom mit Hilfe der XOR-Funktion. Der Schlüsselstrom ist eine pseudozufällige Zeichenfolge. Die meisten Stromchiffrierungen benutzen einen symmetrischen Schlüssel. Der Schlüssel bestimmt den Initialzustand des Systems.

Bevorzugt wird im Rahmen der Erfindung so vorgegangen, dass die Generierung des Schlüsselstroms unter Verwendung wenigstens eines rückgekoppelten Schieberegisters erfolgt, das zu seiner Initialisierung mit einer definierten Bitfolge gefüllt wird. Linear rückgekoppelte Schieberegister können effizient sowohl direkt in Hardware, wie beispielsweise FPGAs, als auch in Software implementiert werden. Rückgekoppelte Schieberegister sind schnell und produzieren Pseudozufallsfolgen mit guten statistischen Eigenschaften. Ein rückgekoppeltes Schieberegister ist in der Digitaltechnik als ein Schieberegister mit n Speicherelementen realisiert. Die einzelnen Speicherelemente sind typischerweise D-Flipflops, welche je ein Bit speichern können. Im Gegensatz zu einem herkömmlichen Schieberegister bestehen zwischen bestimmten

D-Flipflops Abzweigungen, welche die Rückkopplungen darstellen. Zur Rückkoppelung wird in der Regel jeweils eine XOR-Funktion verwendet. Statt der XOR-Verknüpfung kann aber auch eine XNOR-Verknüpfung eingesetzt werden.

Zur Initialisierung kann das Schieberegister mit XOR-Rückkopplung mit beliebigen Werten gefüllt werden, die den vom Schieberegister in der Folge generierten Schlüsselstrom bestimmen. Wie jedes andere Schieberegister verfügt auch das rückgekoppelte Schieberegister über einen Takteingang: Bei jedem Taktimpuls wird in den Folgezustand gewechselt, d.h. wenn ein Bit ausgegeben werden soll, werden alle Bits im Schieberegister um einen Speicherplatz verschoben; das neue Bit am Ende des Schieberegisters wird abhängig von den anderen Bits berechnet. Dieser Vorgang zählt als ein Takt. Für einen vollständigen Durchlauf aller Kombinationen sind 2^n-1 Taktimpulse notwendig. Eine derartige Codesequenz hat somit eine Länge von 2^n-1 bit (n = Anzahl der codegenerierenden in Reihe geschalteten Speicherelemente des Schieberegisters). Als Schlüsselstromgenerator werden in der Regel mehrere lineare rückgekoppelte Schieberegister eingesetzt, die meist unterschiedlich lang sind und unterschiedliche Rückkopplungspolynome haben. Damit kombiniert man lineare rückgekoppelte Schieberegister zu nichtlinearen Generatoren.

Je größer die Länge der Codesequenz des Schlüsselstroms bzw. des Codes ist, desto schwerer ist dieser zu entschlüsseln. Beispielsweise bräuchte ein unendlicher Code gar nicht versteckt zu werden, da er ja nie ganz bekannt ist. Funktionell ist jeder Code als unendlich anzusehen, der sich nicht vor dem Ende der zu verschlüsselnden Information wiederholt. Ein funktionell unendlicher Code hat den Nachteil, dass er nicht übertragen werden kann; er muss generiert werden.

Nachteilig bei Codegeneratoren in der Form von herkömmlichen rückgekoppelten Schiebregistern ist die Tatsache, dass von der Codesequenz leicht auf die Struktur des Generators geschlossen werden kann, so dass sie mit einem gleichgebauten Generator nachgeneriert werden kann. Eine Erhöhung der Sicherheit wird gemäß einer bevorzugten Verfahrensweise im Rahmen der Erfindung dadurch erreicht, dass für jeden Schreibzugriff der Verarbeitungseinheit auf die Datenstruktur oder die Indexstruktur ein anderer Schlüsselstrom verwendet wird. Dies bedeutet, dass das bzw. die rückgekoppelte(n) Schieberegister für die Verschlüsselung jedes Datenpakets neu initialisiert wird bzw. werden. Bevorzugt wird hierbei so vorgegangen, dass zur Initialisierung des bzw. der rückgekoppelten Schieberegister jeweils wenigstens eine erste Bitfolge und eine zweite Bitfolge verwendet wird. Dies erfolgt insbesondere dann, wenn zur Generierung des Schlüsselstroms lediglich ein einziges rückgekoppeltes Schieberegister verwendet wird, derart, dass die erste und die zweite Bitfolge mit Hilfe einer XOR-Funktion verknüpft werden und die sich aus der Verknüpfung ergebende Bitfolge zur Initialisierung dem rückgekoppelten Schieberegister zugeführt wird. Alternativ, und zwar insbesondere für den Fall, dass wenigstens zwei miteinander verschaltete rückgekoppelte Schieberegister für die Generierung des Schlüsselstroms verwendet werden, wird so vorgegangen, dass wenigstens ein erstes rückgekoppeltes Schieberegister zu seiner Initialisierung mit der ersten Bitfolge gefüllt wird und wenigstens ein zweites rückgekoppeltes Schieberegister zu seiner Initialisierung mit der zweiten Bitfolge gefüllt wird.

Im Rahmen der Erfindung muss sichergestellt sein, dass die Verschlüsselung eines Abschnitts des Datenbestandes, wie z.B.

eines Datensatzes der Datenbank und die Entschlüsselung desselben Abschnitts bzw. Datensatzes miteinander synchronisiert sind, d.h. dass die Verschlüsselung und die Entschlüsselung mit dem selben Schlüsselstrom erfolgt. Dies bedeutet, dass der Codegenerator an die Stelle des Verschlüsselungsanfangs gerückt werden muss. Die Synchronisation erfolgt bevorzugt unter Verwendung der Indices der Datensätze. Insbesondere wird so vorgegangen, dass als erste Bitfolge eine dem zu verschlüsselnden oder entschlüsselnden Datensatz zugeordnete Indexnummer gewählt wird oder dass die erste Bitfolge aus dieser generiert wird. Bevorzugt kommt hierbei der Primärindex zum Einsatz. Die zweite Bitfolge ist bevorzugt eine eindeutige Kennung der Datenbank oder wird aus dieser generiert.

Eine noch höhere Sicherheit ergibt sich, wenn, wie dies einer weiteren bevorzugten Verfahrensweise entspricht, zur Initialisierung des bzw. der rückgekoppelten Schieberegister weiters eine dritte Bitfolge verwendet wird. Die dritte Bitfolge ist dabei mit Vorteil eine eindeutige Kennung des jeweiligen Benutzers oder wird aus dieser generiert. Die dritte Bitfolge wird bevorzugt zur Initialisierung einem dritten rückgekoppelten Schieberegister zugeführt.

Ein weiterer Vorteil des erfindungsgemäßen Verfahrens ist, dass die Generierung des Schlüsselstroms schon beginnen kann, sobald wenigstens eines der rückgekoppelten Schieberegister mit dem ersten Bit aus der jeweiligen Bitfolge gefüllt wird. Insbesondere werden die rückgekoppelten Schieberegister gleichzeitig mit der jeweiligen Bitfolge gefüllt.

Die Struktur des Schlüsselstromgenerators ist wie an sich bekannt bevorzugt so, dass zur Rückkoppelung des bzw. der Schieberegister wenigstens ein XOR-Gatter verwendet wird. Die

Komplexität des Generators kann dabei in einfacher Weise dadurch erhöht werden, dass die rückgekoppelten Schieberegister derart miteinander verschaltet sind, dass in Abhängigkeit vom Zustand des einen Schieberegisters das wenigstens eine XOR-Gatter des anderen Schieberegister an- oder abgeschaltet wird.

Eine überaus bevorzugte Weiterbildung ergibt sich, wenn ein Codegenerator zum Einsatz gelangt, wie er in der WO 03/075507 A1 beschrieben ist, wobei auf die Ansprüche 15 und 16 sowie 33 bis 38 der vorliegenden Anmeldung verwiesen wird. Bei einem derartigen Codegenerator kann die Verschlüsselung nicht einmal dann gebrochen werden, wenn sowohl die Struktur des Codegenerators als auch der in ihm ablaufende Algorithmus bekannt sind. Die Struktur des Generators ist nämlich so geartet, dass sie eine derartig hohe Anzahl an unterschiedlichen Codes in einer derartig großen Länge zu generieren im Stande ist, dass die Entdeckung des gerade verwendeten Codes so wie die aktuell produzierte Stelle in der Codesequenz nur mit einer extrem geringen Wahrscheinlichkeit möglich ist. Der Code kann dann nicht nachgeneriert werden, wenn der Generator so viele verschiedene Codes erstellen kann, dass von einem Abschnitt des einzelnen Codes nicht auf dessen Fortsetzung geschlossen werden kann.

Der Zugriff eines Benutzerrechners auf den Datenbestand bzw. die Datenbank erfolgt in der Regel von einem entfernten Ort über eine Datenkommunikationsverbindung, insbesondere über ein Computernetzwerk. Der Zugriff eines Benutzerrechners auf die Datenstruktur und die Indexstruktur erfolgt hierbei über die Verarbeitungseinheit. Da die Daten in der Verarbeitungseinheit im Klartext vorliegen, ist es vorteilhaft, Vorkehrungen zu schaffen, um zu verhindern, dass Benutzerrechner

Zugriff auf diese Klartextdaten erlangen. Die Erfindung sieht in diesem Zusammenhang bevorzugt vor, dass die zwischen der Verarbeitungseinheit und einem Benutzerrechner übermittelten Daten verschlüsselt übermittelt werden. Insbesondere wird so vorgegangen, dass die verschlüsselte Übermittlung der Daten zwischen der Verarbeitungseinheit und dem Benutzerrechner unter Verwendung jeweils einer dem Benutzerrechner und einer dem Datenbestand zugeordneten Ver- und Entschlüsselungseinheit erfolgt, mit der die Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselt werden.

Eine besonders sichere Ausführung wird dadurch sichergestellt, dass jegliche Übermittlung von Daten von und zu der Verarbeitungseinheit über wenigstens eine Ver- und Entschlüsselungseinheit erfolgt, mit der die Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselt werden. Die Verarbeitungseinheit verfügt somit über keinen unverschlüsselten Eingang oder Ausgang in/aus das/dem umgebende/n Netzwerk, sodass sichergestellt ist, dass die Dateispeicher der Verarbeitungseinheit, in denen die Daten im Klartext vorliegen, nicht eingesehen werden können.

Gemäß einem weiteren Aspekt der vorliegenden Erfindung wird eine Vorrichtung der eingangs genannten Art zum Schreiben und Lesen von Daten in einen bzw. aus einem indizierten Datenbestand vorgeschlagen. Die erfindungsgemäße Vorrichtung zeichnet sich dadurch aus, dass die Verarbeitungseinheit mit der Datenstruktur und mit der Indexstruktur über mindestens eine Ver- und Entschlüsselungseinheit verbunden ist, mit der die Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselbar sind, sodass der Schreib-/Lesezugriff der Verarbeitungseinheit auf die Indexstruktur und auf die

Datenstruktur über die mindestens eine Ver- und Entschlüsselungseinheit erfolgt.

Die Verarbeitungseinheit ist bevorzugt als eine der Datenstruktur und der Indexstruktur zugeordnete CPU ausgebildet.

Bevorzugte Weiterbildungen der erfindungsgemäßen Vorrichtung ergeben sich aus den Unteransprüchen.

Die Erfindung wird in der Folge anhand von in der Zeichnung schematisch dargestellten Ausführungsbeispielen näher erläutert. In dieser zeigen Fig.1 eine erfindungsgemäße Datenbank, Fig. 2 den Verschlüsselungs- und Entschlüsselungsvorgang und Fig. 3, Fig. 4 und Fig. 5 verschiedene Ausbildungen eines im Rahmen der Erfindung verwendeten Schlüsselstromgenerators.

In Fig. 1 ist eine Datenbank 1 dargestellt, die eine Datenstruktur 2 und eine Indexstruktur 3 umfasst. Mit 4 ist eine Verarbeitungseinheit in der Form einer CPU bezeichnet, welche eine Schnittstelle 5 für aus- und eintreffende Daten aufweist und die den Schreib- und den Lesezugriff auf die Indexstruktur 3 und die Datenstruktur 2 steuert. Die Verarbeitungseinheit 4 ist über eine Ver- und Entschlüsselungseinheit 6 mit der Datenstruktur 2 und über eine Ver- und Entschlüsselungseinheit 7 mit der Indexstruktur 3 verbunden, sodass der Schreib- bzw. Lesezugriff auf die Datenstruktur 2 und die Indexstruktur 3 über die Ver- und Entschlüsselungseinheit 6 bzw. 7 erfolgt.

In der Verarbeitungseinheit 4 liegen die in die Datenstruktur 2 oder die Indexstruktur 3 zu schreibenden und die aus der Datenstruktur 2 oder der Indexstruktur 3 auszulesenden Daten im Klartext vor, sodass die für das Indizieren und für das

Auffinden von Datensätzen erforderlichen Operationen durchgeführt werden können. In der Datenstruktur 2 und in der Indexstruktur 3 liegen die Daten hingegen ausschließlich in verschlüsselter Form vor. Damit die Verarbeitungseinheit 4 auf die verschlüsselten Daten zugreifen kann, erfolgt der Zugriff über die Ver- und Entschlüsselungseinheit 6 bzw. 7. In Fig. 1 ist die Übertragung der Daten im Klartext hierbei mit durchgezogener Linie und die Übertragung von verschlüsselten Daten mit strichlierter Linie dargestellt.

Die Ver- und Entschlüsselungseinheiten 6 und 7 ver- und entschlüsseln die jeweiligen Daten mittels einer Stromchiffrierung und umfassen dementsprechend einen Schlüsselstromgenerator, der anhand der verschiedenen Ausführungsformen in Fig. 2 bis 5 näher erläutert wird. Jede der nachfolgend beschriebenen Ausführungsformen kann im Rahmen der Ver- und Entschlüsselungseinheit 6 bzw. 7 oder der Ver- und Entschlüsselungseinheit 10 bzw. 11 (siehe unten) zum Einsatz gelangen.

Ein Benutzerrechner 8 kann über eine Kommunikationsverbindung 9 auf die Datenbank 1 zugreifen. Die Daten werden über die Kommunikationsverbindung 9 in verschlüsselter Form übertragen, wobei die Ver- und die Entschlüsselung mittels der Ver- und Entschlüsselungseinheiten 10 und 11 erfolgt. Die Ver- und die Entschlüsselung erfolgt bevorzugt mittels einer Stromchiffrierung.

Die Ver- und Entschlüsselungseinheiten 6, 7, 10 und 11 können jeweils einen Codegenerator gemäß WO 03/075507 A1 umfassen, wobei die Codegeneratoren der Ver- und Entschlüsselungseinheiten 6 und 7 für die Verschlüsselung und die spätere Entschlüsselung von Daten synchronisiert werden müssen. Weiters

müssen die Codegeneratoren der Ver- und Entschlüsselungseinheiten 10 und 11 miteinander synchronisiert sein.

Wenn nun ein Benutzer eine bestimmte Information unter den von ihm in der Datenbank 1 abgelegten Daten sucht, gibt er entsprechende Suchworte in den Benutzerrechner 8 ein. Dieser verschlüsselt diese Eingabe und übermittelt sie an die Datenbank 1. In der Datenbank 1 wird dieser Suchbegriff mittels der Ver- und Entschlüsselungseinheit 11 entschlüsselt und der Verarbeitungseinheit 4 im Klartext zur Verfügung gestellt. Die Verarbeitungseinheit 4 sucht den Suchbegriff in der Indexstruktur 3, wobei die Indexstruktur 3 der Verarbeitungseinheit 4 auf Grund des Echtzeit-Zugriffs über die Ver- und Entschlüsselungseinheit 7 klartextlich zur Verfügung steht. Die Indexstruktur 3 gibt den genauen Standort der gesuchten Daten in der verschlüsselten Datenstruktur 2 bekannt. Daraufhin werden die verschlüsselten Daten in der Datenstruktur 2 aufgesucht und in unverändert verschlüsselter Form an den Benutzerrechner 8 des Benutzers weitergeleitet. Der Benutzerrechner 8 entschlüsselt die Daten mit Hilfe der Ver- und Entschlüsselungseinheit 10, sodass diese als angeforderte Klartextdaten dort angezeigt werden. Alternativ kann das Auslesen der verschlüsselten Daten aus der Datenstruktur 2 über die Ver- und Entschlüsselungseinheit 6 erfolgen, wobei diese Daten in der Verarbeitungseinheit 4 dann im Klartext vorliegen und zwecks Übertragung an den Benutzerrechner 8 mit der Ver- und Entschlüsselungseinheit 11 wieder verschlüsselt werden müssen.

Die Synchronisierung der Ver- und Entschlüsselungseinheiten 6 und 7 wird anhand der Fig. 2 näher erläutert. Fig. 2 zeigt eine Prinzipschaltung eines Schlüsselstromgenerators 12 mit einem Schieberegister 13, das aus einer Mehrzahl von zu einer

codeproduzierenden Reihe zusammengeschalteten Speicherelementen, nämlich Flip-Flops FF1, FF2, ... FF9 besteht. Ein XOR-Gatter XORp1 ist so verschaltet, dass der eine Eingang des XOR-Gatters XORp1 mit dem Ausgang des in der codeproduzierenden Reihe befindlichen Speicherelements FF2 und der andere Eingang des XOR-Gatters XORp1 mit dem Ausgang des in der codeproduzierenden Reihe befindlichen Speicherelements FF5 und der Ausgang des XOR-Gatters XORp1 mit dem Eingang des in Flussrichtung mit dem einen Eingang des XOR-Gatters XORp1 verbundenen Speicherelements FF2 in der Reihe nachfolgenden Speicherelements FF3 - sohin rekursiv - verbunden ist. Weiters ist ersichtlich, dass das letzte Speicherelement FF9 über einen Inverter INV mit dem ersten Speicherelement FF1 verbunden ist. Sobald man das Schieberegister 13 mit einer Bitfolge befüllt, erhält man mit dieser Schaltung eine Codesequenz. Wenn, wie dies bei der Ausbildung gemäß Fig. 2 der Fall ist, nur ein einziges Schieberegister zum Einsatz gelangt, werden die Bitfolgen 14, 15 und 16 dem Schieberegister 13 zu dessen Initialisierung derart zugeführt, dass zunächst die Bitfolgen 14 und 15 mit Hilfe eines XOR-Gatters 17 miteinander verknüpft werden und dann die verknüpfte Bitfolge mit der Bitfolge 16 mit Hilfe des XOR-Gatters 18 verknüpft wird. Dabei ist es bevorzugt, dass die aus den Bitfolgen 14, 15 und 16 generierte, dem Schieberegister 13 zugeführte Bitfolge nicht länger ist als dies der Anzahl der Speicherelemente im Schieberegister 13 entspricht, da die Bitfolge andernfalls von der über den Inverter INV aus dem Speicherelemente FF9 kommenden Bitfolge überlagert würde. Die erste Bitfolge 14 entspricht dabei der Indexnummer des betreffenden Datensatzes. Die zweite Bitfolge 15 entspricht dabei der Datenbank ID. Die dritte Bitfolge 16 entspricht der "Own ID" des Benutzers.

Der Schlüsselstromgenerator 12 erzeugt einen Schlüsselstrom 19a. Ein eingehender Strom 19b von Klartextdaten wird so verschlüsselt, dass die Bits des Bitstroms 19b des Klartextes mit den Bits eines Schlüsselstroms 19a einzeln mit Hilfe eines XOR-Gatters 20 verknüpft werden. Wenn die Klartextdaten einen Datensatz der Datenbank 1 repräsentieren, wird der Index dieses Datensatzes gemäß der der Datenbank innewohnenden Strukturierung ermittelt und als Bitfolge 14 dem Schlüsselstromgenerator 12 als Initialisierungssequenz zugeführt.

Wenn der Strom 19b ein Strom von verschlüsselten Daten ist, wird dieser so entschlüsselt, dass die Bits des Bitstroms 19b der verschlüsselten Daten mit den Bits des Schlüsselstroms 19a einzeln mit Hilfe eines XOR-Gatters 20 verknüpft werden. Wenn die verschlüsselten Daten einen verschlüsselten Datensatz der Datenbank 1 repräsentieren, wird der Index dieses Datensatzes gemäß der der Datenbank innewohnenden Strukturierung ermittelt und als Bitfolge 14 dem Schlüsselstromgenerator 12 als Initialisierungssequenz zugeführt.

Bei der abgewandelten Ausbildung gemäß Fig. 3 gelangen insgesamt drei Schieberegister 21, 22 und 23 zum Einsatz. Die Speicherelemente der einzelnen Schieberegister sind in diesem Beispiel jeweils auf gleiche Weise rekursiv verschaltet wie in Fig. 2. Die Schieberegister sind weiters derart miteinander verschaltet, dass in Abhängigkeit vom Zustand des zweiten Schieberegisters 22 die Funktion des XOR-Gatters XORp1 der rekursiven Verschaltung des ersten Schieberegisters 21 an- und abgeschaltet wird. Die Funktion des XOR-Gatters XORpp1 der rekursiven Verschaltung des zweiten Schieberegisters 22 wird wiederum in Abhängigkeit vom Zustand des dritten Schieberegisters 23 an- und abgeschaltet. Zu diesem Zweck ist der

Ausgang des Flip-Flops FFp2 bzw. FFpp2 des einen Schiebregisters 22 bzw. 23 mit dem Eingang eines UND-Gatters UNDp1 bzw. UNDpp1 verbunden, das in die jeweilige rekursive Funktion XORp1 bzw. XORpp1 der Schieberegister 21 bzw. 22 eingefügt ist.

Es entsteht somit ein Codegenerator 12 mit drei Ebenen, wobei die Codegenerierung auf jeder Ebene durch Initialisieren des jeweiligen Schieberegisters 21, 22 und 23 mit der Bitfolge 14, 15 und 16 beeinflusst wird. Die Initialisierung kann dabei bevorzugt so erfolgen, dass dem Schieberegister 21 der ersten Ebene die erste Bitfolge 14, dem Schieberegister 22 der zweiten Ebene die zweite Bitfolge 15 und dem Schieberegister 23 der dritten Ebene die dritte Bitfolge 16 zugeführt wird, wobei die Bitfolgen 14, 15 und 16 bevorzugt so definiert sind wie in Fig. 2 beschrieben.

Bei der Ausbildung gemäß Fig. 4 ist die in Fig. 3 gezeigte Struktur noch komplexer ausgestaltet und es sind insbesondere längere codeproduzierende Reihen und eine Mehrzahl von rekursiven Verschaltungen vorgesehen. Dabei ist eine Anzahl ununterbrochener in Reihe geschalteter Speicherelemente in Form von Schieberegister SRG1, SRG2,... verwirklicht, die funktionell gesehen gemeinsam ein Schieberegister 24 im Sinne der Erfindung bilden. Es verdoppelt sich die Länge des Codes pro hinzugefügtem Speicherelement, so dass sich die Länge des Codes wie folgt berechnet

$$Lc = 2^n-1$$

(Lc = Länge der Codesequenz; n = Anzahl der codegenerierenden in Reihe geschalteten Speicherelemente)

Wenn diese Einheit mit einem bestimmten Takt betrieben wird gilt für die Dauer des Codes:

$$Tc = \frac{2^n - 1}{fc}$$

(Tc = Dauer bis sich der Code wiederholt; fc = Codegenerierungstaktfrequenz)

Mit weniger als 50 Speicherelementen bei einer Codegenerierungstaktfrequenz von 384.000 Bit/s läuft der Code länger als ein Jahr ohne dass sich die Sequenz wiederholt, so dass ein zu verschlüsselndes Signal simultan über einen ebenso langen Zeitraum verschlüsselt über eine Standleitung übersendet und entschlüsselt werden kann, so dass Übertragungen live über einen ebenso langen Zeitraum möglich sind.

Wenn man nun bei entsprechender Länge des Schieberegisters 24 an mehren Stellen dieses Schieberegisters 24 zwischen einem Speicherelement FF1,2,3,4 und dem nächsten in der Reihe befindlichen Speicherelement FF2,3,4,5 ein XOR-Gatter XORp1,p2,p3,p4 einfügt und dieses dann mit dem Signal von einem dritten Speicherelement FF8,15,20,23 speist, so verändert man jeweils den dadurch erzeugten Code (Fig. 5).

Bei einer Mehrzahl von codeverändernden XOR-Gattern XORp1,p2,p3,p4, siehe Fig. 5, soll sichergestellt sein, dass die verschiedenen codeverändernden XOR-Gatter XORp1,p2,p3,p4, deren erster Eingang von einem Ausgang eines Speicherelements FF1,2,3,4 gespeist wird, ihren zweiten Eingang jeweils vom Ausgang eines Speicherelements FF8,15,20,23 gespeist erhalten, welches eine Anzahl von Speicherelementen in Flussrichtung vom erstgenannten Speicherelement FF1,2,3,4 entfernt

ist, welche jeweils einer unterschiedlichen Primzahl entspricht, die größer als 1 aber kein Teilbetrag der Gesamtzahl der in Reihe R geschalteten Speicherelemente ist, sodass es bei der Beeinflussung der Codesequenz zu keinen codesequenzverkürzenden Resonanzeffekten kommt. Zwischen den entsprechenden Speicherelementpaaren FF1,8; FF2,15; FF3,20; FF4,23 liegt also jeweils eine Anzahl von 7, 13, 17 und 19 (Primzahlen) Speicherelementen.

Wenn man an einen der beiden Eingänge des jeweiligen XOR-Gatters XORp1 bzw. XORp1,p2,p3,p4 den Ausgang eines UND-Gatters UNDp1 bzw. UNDp1,p2,p3,p4 dessen einer Eingang am Ausgang des Speicherelements FF3 bzw. FF8,15,20,23 hängt, anschließt, dann kann man dieses XOR-Gatter XORp1 bzw. XORp1,p2,p3,p4 in seiner codeverändernden Wirkung über den zweiten Eingang des UND-Gatters UNDp1 bzw. UNDp1,p2,p3,p4 an- und abschalten und wenn man daran jeweils ein weiteres Speicherelement FFp1 bzw. FFp1,p2,p3,p4 anschließt, das An- und Abschalten der codebeeinflussenden Wirkung des XOR-Gatters XORp1 bzw. XORp1,p2,p3,p4 programmierbar machen. Die codeprogrammierenden Speicherelemente FFp1,p2,p3,p4 können dabei zu einem Schieberegister 25 zusammengeschaltet sein. In weiterer Folge können die codeprogrammierenden Speicherelemente FFp1,p2,p3,p4 des Schieberegisters 25 selbst wiederum mit Hilfe eines XOR-Gatters XORpp1 rekursiv verschaltet werden.

Die Anzahl der programmierbaren unterschiedlichen Codes berechnet sich wie folgt:

$$Nc = 2^{pn} - 1$$

(Nc = Anzahl der möglichen unterschiedlichen Codes;
pn = Anzahl der programmierbaren XOR - Gatter

XORp1,p2,...pn)

Wenn man nun im Besitz eines identen Codegenerators ist, und an Hand einer bestimmten Anzahl von Bits den weiteren Verlauf der Codesequenz erschließen möchte so hängt die Wahrscheinlichkeit, mit der man die richtige Fortsetzung der Codesequenz erkennt, sowohl von der Anzahl der in der Codegenerierung verwendeten Speicherelemente FF1,2,...n als auch jener der programmierbaren, codeverändernden XOR-Gatter XORp1,p2,...pn ab. Daraus ergibt sich eine Wahrscheinlichkeit, die dem Code zugrunde liegende Programmierung zu entdecken und sohin den weiteren Verlauf des Codes vorauszusagen von:

$$W = \frac{Nb}{(2^{n} - 1) * (2^{pn} - 1)}$$

(Nb = Anzahl der beobachteten Bits der Codesequenz; n = Anzahl der codegenerierenden in Reihe geschalteten Speicherelemente FF1,2,...n; pn = Anzahl der programmierbar den Code verändernden XOR-Gatter XORp1,p2,...pn)

Beispiel:

233 ist die 52. Primzahl. Wenn man die 1 nicht nützt und die 233 die Gesamtzahl der in Reihe geschalteten Speicherelemente ausdrückt, so befinden sich auf dieser Strecke 50 unterschiedliche Speicherelemente, welche sich jeweils in Entfernung von einem Ausgangs-Speicherelement befinden, die einer Primzahl entspricht (np = 50). Da jedes rekursive XOR-Gatter 1-50 jeweils zwischen einem nächsten Speicherelement 1-50

beginnend vom ersten in Reihe eingeschaltet ist, verlängert sich die Gesamtlänge der Speicherelemente auf (n = 233 + 50 = 283).

Daraus folgt:

$$W = \frac{Nb}{(2^n - 1) * (2^{pn} - 1)} = \frac{Nb}{(2^{283} - 1) * (2^{50} - 1)}$$

$$W = \frac{Nb}{(1{,}5541351138 * 10^{85} - 1) * (1{,}1258999068 * 10^{15} - 1)}$$

$$W \sim \frac{Nb}{1{,}7498005798 * 10^{100}}$$

Mit anderen Worten muss man die Codesequenz $1{,}7498005798 * 10^{100}$ Taktschritte lang beobachten, damit man mit der Wahrscheinlichkeit 1 eine bestimmte Sequenz entdeckt. Wenn die Taktfrequenz 384000 Hz beträgt ergibt dies eine notwendige Beobachtungszeit von $1{,}4449430312*10^{87}$ Jahren.

Indem man die codeprogrammierenden Speicherelemente (FFp1,p2,p3,p4,p5,p6) des Schieberegisters 25 rekursiv miteinander verschaltet, so dass sie innerhalb des Zeitintervalls

$$T\ pn = \frac{2^{pn} - 1}{fp}$$

(T pn = Durchlaufzeit aller möglichen Programmierzustände; pn = Anzahl der Programm-Speicherelemente; fp = Programmiertaktfrequenz)

sämtliche mögliche Zustandskombinationen durchlaufen, ergibt sich die Programmierung aus einer bestimmten Zeitspanne, in der die codeprogrammierenden Speicherelemente mit einem Programmtakt versorgt werden.

Damit aus der Programmierdauer auch nicht annähernd die Programmierung erschließbar ist kann die Programmierung zweistufig erfolgen. Hierzu kann eine weitere Programmierungs-Ebene hinzugefügt werden, indem das codeprogrammierende XOR-Gatter XORppl selbst wiederum unter Zwischenschaltung eines UND-Gatters UNDppl mit einer Speicherelemente-Reihe RRR verbunden und somit programmierbar gemacht wird, wobei wiederum ein XOR-Gatter XORpppl zur rekursiven Verschaltung des Schieberegisters 26 verwendet wird (Fig.6).

Ausgehend von obigem Rechenbeispiel wird dadurch gewährleistet, dass die $(2^{283}-1)*(2^{50}-1)$ verschiedenen Zustände in $2^{50}-1$ verschiedene Abschnitte zergliedert werden, von welchen einer in der ersten Programmierphase ausgewählt wird. Dieser Auswahlvorgang erfolgt in maximal $2^{ppn} - 1$ Schritten (ppn = Anzahl der Primzahlen, die in der Anzahl der bei der Programmierung verwendeten Primzahlen (50) enthalten sind, also 16). Dies bedeutet, dass maximal 2^{16} Schritte erfolgen müssen, ehe sämtliche Abschnitte aufgesucht sind. Bei einer Programmiertaktfrequenz von 1 MHz ist dieser Vorgang in 0,065 Sekunden abgeschlossen. Ein Zeitraum, der wohl bei jeder Programmierung durchmessen wird, da er unter der Reaktionszeit des Menschen liegt, weshalb gewährleistet ist, dass aus der tatsächlich verstrichenen Programmierzeit keine Rückschlüsse auf die Programmierung der Schlüssel gezogen werden können.

Patentansprüche:

1. Verfahren zum Schreiben und Lesen von Daten in einen bzw. aus einem indizierten Datenbestand (1), der eine Datenstruktur (2) und eine zugehörige Indexstruktur (3) umfasst, wobei eine Verarbeitungseinheit (4) zu schreibende Daten im Klartext empfängt und mittels eines Schreibzugriffs in die Datenstruktur (2) schreibt und Indexdaten in der Indexstruktur (3) aktualisiert und wobei die Verarbeitungseinheit (4) auszulesende Daten oder deren Speicherort mittels eines Zugriffs auf die Indexdaten (3) ermittelt und die auszulesenden Daten mittels eines Lesezugriffs aus der Datenstruktur (2) ausliest und im Klartext zur Verfügung stellt, dadurch gekennzeichnet, dass die Daten in der Datenstruktur (2) und die Indexdaten in der Indexstruktur (3) verschlüsselt gespeichert werden und dass der Schreib-/Lesezugriff der Verarbeitungseinheit (4) auf die Indexstruktur (3) und auf die Datenstruktur (2) über mindestens eine Ver- und Entschlüsselungseinheit (6, 7) erfolgt, mit der die Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselt werden.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Generierung des Schlüsselstroms unter Verwendung wenigstens eines rückgekoppelten Schieberegisters (13; 21, 22, 23; 24, 25; 24, 25, 26) erfolgt, das zu seiner Initialisierung mit einer definierten Bitfolge gefüllt wird.

3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass für jeden Schreibzugriff ein anderer Schlüsselstrom verwendet wird.

4. Verfahren nach Anspruch 1, 2 oder 3, dadurch gekennzeichnet, dass zur Initialisierung des bzw. der rückgekoppelten Schieberegister (13; 21, 22, 23; 24, 25; 24, 25, 26) jeweils wenigstens eine erste Bitfolge (14) und eine zweite Bitfolge (15) verwendet wird.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, dass die erste und die zweite Bitfolge (14, 15) mit Hilfe einer XOR-Funktion (17) verknüpft werden und die sich aus der Verknüpfung ergebende Bitfolge zur Initialisierung dem rückgekoppelten Schieberegister (13) zugeführt wird.

6. Verfahren nach Anspruch 4 oder 5, dadurch gekennzeichnet, dass wenigstens ein erstes rückgekoppeltes Schieberegister (21, 24) zu seiner Initialisierung mit der ersten Bitfolge (14) gefüllt wird und wenigstens ein zweites rückgekoppeltes Schieberegister (22, 25) zu seiner Initialisierung mit der zweiten Bitfolge (15) gefüllt wird.

7. Verfahren nach Anspruch 4, 5 oder 6, dadurch gekennzeichnet, dass als erste Bitfolge (14) eine dem zu verschlüsselnden oder entschlüsselnden Datensatz zugeordnete Indexnummer gewählt wird.

8. Verfahren nach einem der Ansprüche 4 bis 7, dadurch gekennzeichnet, dass die zweite Bitfolge (15) aus einer eindeutigen Kennung der Datenbank generiert wird.

9. Verfahren nach einem der Ansprüche 4 bis 8, dadurch gekennzeichnet, dass zur Initialisierung des bzw. der rückgekoppelten Schieberegister (13; 21, 22, 23; 24, 25; 24, 25, 26) weiters eine dritte Bitfolge (16) verwendet wird.

10. Verfahren nach Anspruch 9, dadurch gekennzeichnet, dass die dritte Bitfolge (16) aus einer eindeutigen Kennung des jeweiligen Benutzers generiert wird.

11. Verfahren nach Anspruch 9 oder 10, dadurch gekennzeichnet, dass die dritte Bitfolge (16) zur Initialisierung einem dritten rückgekoppelten Schieberegister (23, 26) zugeführt wird.

12. Verfahren nach einem der Ansprüche 4 bis 11, dadurch gekennzeichnet, dass die rückgekoppelten Schieberegister (13; 21, 22, 23; 24, 25; 24, 25, 26) gleichzeitig mit der jeweiligen Bitfolge (14, 15, 16) gefüllt werden.

13. Verfahren nach einem der Ansprüche 2 bis 12, dadurch gekennzeichnet, dass zur Rückkoppelung des bzw. der Schieberegister (13; 21, 22, 23; 24, 25; 24, 25, 26) wenigstens ein XOR-Gatter (XORp1, XORp2, XORp3, XORp4, XORpp1, XORppp1) verwendet wird.

14. Verfahren nach einem der Ansprüche 2 bis 13, dadurch gekennzeichnet, dass die rückgekoppelten Schieberegister (13; 21, 22, 23; 24, 25; 24, 25, 26) derart miteinander verschaltet sind, dass in Abhängigkeit vom Zustand des einen Schieberegisters das wenigstens eine XOR-Gatter (XORp1, XORp2, XORp3, XORp4, XORpp1, XORppp1) des anderen Schieberegister an- oder abgeschaltet wird.

15. Verfahren nach einem der Ansprüche 2 bis 14, dadurch gekennzeichnet, dass das wenigstens eine rückgekoppelte Schieberegister (13; 21, 22, 23; 24, 25; 24, 25, 26) eine Mehrzahl von zu einer codeproduzierenden Reihe geschalteten Speicherelementen (FF1, FF2,...; FFp1, FFp2,...; FFpp1,

FFpp2,...) aufweist, wobei der Ausgang des in der Reihe letzten Speicherelements mit dem Eingang des in der Reihe ersten Speicherelements zu einem Kreis zusammengeschlossen ist, wobei die Rückkoppelung mit Hilfe des wenigstens einen XOR-Gatters (XORp1, XORp2, XORp3, XORp4, XORpp1, XORppp1) derart erfolgt, dass der erster Eingang des XOR-Gatters mit dem Ausgang eines in der codeproduzierenden Reihe befindlichen Speicherelements (FF2), der zweite Eingang mit dem Ausgang eines weiteren in der codeproduzierenden Reihe befindlichen Speicherelements (FF5) und der Ausgang mit dem Eingang des in der codeproduzierenden Reihe dem mit dem ersten Eingang des XOR-Gatters verbundenen Speicherelement nachfolgenden Speicherelements (FF3) verbunden ist.

16. Verfahren nach Anspruch 15, dadurch gekennzeichnet, dass in die den zweiten Eingang des wenigstens einen XOR-Gatters (XORp1) und den Ausgang des weiteren in der codeproduzierenden Reihe (21, 24) befindlichen Speicherelements (FF5) verbindende Leitung ein UND-Gatter (UNDp1) derart geschalten ist, dass der Ausgang des UND-Gatters (UNDp1) mit dem zweiten Eingang des XOR-Gatters(XORp1), der erste Eingang des UND-Gatters (UNDp1) mit dem Ausgang des weiteren in der codeproduzierenden Reihe befindlichen Speicherelements (FF5) und der zweite Eingang des UND-Gatters (UNDp1) mit dem Ausgang eines codeprogrammierenden Speicherelements (FFp2) verbunden ist, wobei als codeprogrammierendes Speicherelement ein Speicherelement eines weiteren rückgekoppelten Schieberegisters (22, 25) verwendet wird, und dass bevorzugt der Ausgang eines in der codeproduzierenden Reihe (21, 24) befindlichen Speicherelements (FF9) mit dem Eingang eines Inverters (INV) und der Ausgang des Inverters (INV) mit dem Eingang eines anderen in der codeproduzierenden Reihe (21, 24) angeordneten Speicherelements (FF1) verbunden ist.

17. Verfahren nach einem der Ansprüche 1 bis 16, dadurch gekennzeichnet, dass der Datenbestand (1) eine Datenbank ist.

18. Verfahren nach einem der Ansprüche 1 bis 17, dadurch gekennzeichnet, dass die zwischen der Verarbeitungseinheit (4) und einem Benutzerrechner (8) übermittelten Daten verschlüsselt übermittelt werden.

19. Verfahren nach Anspruch 18, dadurch gekennzeichnet, dass die verschlüsselte Übermittlung der Daten zwischen der Verarbeitungseinheit (4) und dem Benutzerrechner (8) unter Verwendung jeweils einer dem Benutzerrechner (8) und einer dem Datenbestand (1) zugeordneten Ver- und Entschlüsselungseinheit (11) erfolgt, mit der die Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselt werden.

20. Verfahren nach einem der Ansprüche 1 bis 19, dadurch gekennzeichnet, dass jegliche Übermittlung von Daten von und zu der Verarbeitungseinheit (4) über wenigstens eine Ver- und Entschlüsselungseinheit (6, 7, 11) erfolgt, mit der die Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselt werden.

21. Vorrichtung zum Schreiben und Lesen von Daten in einen bzw. aus einem indizierten Datenbestand (1), der eine Datenstruktur (2) und eine zugehörige Indexstruktur (3) umfasst, umfassend eine Verarbeitungseinheit (4), in der zu schreibende Daten im Klartext empfangen werden können und die einen Schreibzugriff auf die Datenstruktur (2) aufweist, um die Daten in die Datenstruktur (2) zu schreiben, und die mit der Indexstruktur (3) zusammenwirkt, um Indexdaten in der Indexstruktur (3) zu aktualisieren, und die einen Zugriff auf die

Indexdaten aufweist, um auszulesende Daten oder deren Speicherort zu ermitteln, und die einen Lesezugriff auf die Datenstruktur (2) aufweist, um die auszulesenden Daten aus der Datenstruktur (2) auszulesen und im Klartext zur Verfügung zu stellen, dadurch gekennzeichnet, dass die Verarbeitungseinheit (4) mit der Datenstruktur (2) und mit der Indexstruktur (3) über mindestens eine Ver- und Entschlüsselungseinheit (6,7) verbunden ist, mit der die Daten mittels einer Stromchiffrierung ver- bzw. entschlüsselbar sind, sodass der Schreib-/Lesezugriff der Verarbeitungseinheit (4) auf die Indexstruktur (3) und auf die Datenstruktur (2) über die mindestens eine Ver- und Entschlüsselungseinheit (6,7) erfolgt.

22. Vorrichtung nach Anspruch 21, dadurch gekennzeichnet, dass die Ver- und Entschlüsselungseinheit (6,7) zur Generierung eines Schlüsselstroms wenigstens ein rückgekoppeltes Schieberegister (13; 21,22,23; 24,25; 24,25,26) aufweist, dem zu seiner Initialisierung jeweils eine definierte Bitfolge zugeführt ist.

23. Vorrichtung nach Anspruch 22, dadurch gekennzeichnet, dass Mittel zum Generieren und/oder Speichern wenigstens einer ersten Bitfolge (14) und einer zweiten Bitfolge (15) vorgesehen sind, die mit dem bzw. den Schieberegister(n) (13; 21,22,23; 24,25; 24,25,26) derart zusammenwirken, dass wenigstens die erste Bitfolge (14) und die zweite Bitfolge (15) zur Initialisierung des bzw. der rückgekoppelten Schieberegister (13; 21,22,23; 24,25; 24,25,26) verwendet werden.

24. Vorrichtung nach Anspruch 23, dadurch gekennzeichnet, dass die erste Bitfolge (14) wenigstens einem ersten rückgekoppelten Schieberegister (21; 24) zu dessen Initialisierung zugeführt ist und die zweite Bitfolge (15) wenigstens einem

zweiten rückgekoppelten Schieberegister (22; 25) zu dessen Initialisierung zugeführt ist.

25. Vorrichtung nach Anspruch 23 oder 24, dadurch gekennzeichnet, dass die Mittel zum Generieren und/oder Speichern der ersten Bitfolge (14) ausgebildet sind, um die erste Bitfolge (14) aus einer dem zu verschlüsselnden oder entschlüsselnden Datensatz zugeordneten Indexnummer zu generieren.

26. Vorrichtung nach einem der Ansprüche 23 bis 25, dadurch gekennzeichnet, dass die Mittel zum Generieren und/oder Speichern der zweiten Bitfolge (15) ausgebildet sind, um die zweite Bitfolge (15) aus einer eindeutigen Kennung der Datenbank (1) zu generieren.

27. Vorrichtung nach einem der Ansprüche 23 bis 26, dadurch gekennzeichnet, dass Mittel zum Generieren und/oder Speichern wenigstens einer dritten Bitfolge (16) vorgesehen sind, die mit dem bzw. den Schieberegister(n) (13; 21,22,23; 24,25; 24,25,26) derart zusammenwirken, dass auch die dritte Bitfolge (16) zur Initialisierung des bzw. der rückgekoppelten Schieberegister (13; 21,22,23; 24,25; 24,25,26) verwendet wird.

28. Vorrichtung nach Anspruch 27, dadurch gekennzeichnet, dass die dritte Bitfolge (16) aus einer eindeutigen Kennung des jeweiligen Benutzers generiert wird.

29. Vorrichtung nach Anspruch 27 oder 28, dadurch gekennzeichnet, dass die dritte Bitfolge (16) zur Initialisierung einem dritten rückgekoppelten Schieberegister (23,26) zugeführt ist.

30. Vorrichtung nach einem der Ansprüche 23 bis 29, dadurch gekennzeichnet, dass die rückgekoppelten Schieberegister (13; 21,22,23; 24,25; 24,25,26) gleichzeitig mit der jeweiligen Bitfolge gefüllt werden.

31. Vorrichtung nach einem der Ansprüche 22 bis 30, dadurch gekennzeichnet, dass zur Rückkoppelung des bzw. der Schieberegister (13; 21,22,23; 24,25; 24,25,26) wenigstens ein XOR-Gatter (XORp1, XORp2, XORp3, XORp4, XORpp1, XORppp1) eingesetzt ist.

32. Vorrichtung nach Anspruch 31, dadurch gekennzeichnet, dass die rückgekoppelten Schieberegister (13; 21,22,23; 24,25; 24,25,26) derart miteinander verschaltet sind, dass in Abhängigkeit vom Zustand des einen Schieberegisters das wenigstens eine XOR-Gatter (XORp1, XORp2, XORp3, XORp4, XORpp1) des anderen Schieberegister an- oder abgeschaltet wird.

33. Vorrichtung nach Anspruch 31 oder 32, dadurch gekennzeichnet, dass das wenigstens eine rückgekoppelte Schieberegister (13; 21,22,23; 24,25; 24,25,26) eine Mehrzahl von zu einer codeproduzierenden Reihe geschalteten Speicherelementen (FF1, FF2,...; FFp1, FFp2,...; FFpp1, FFpp2,...) aufweist, wobei der Ausgang des in der Reihe letzten Speicherelements mit dem Eingang des in der Reihe ersten Speicherelements zu einem Kreis zusammengeschlossen ist, wobei die Rückkoppelung mit Hilfe des wenigstens einen XOR-Gatters (XORp1, XORp2, XORp3, XORp4, XORpp1, XORppp1) derart erfolgt, dass der erster Eingang des XOR-Gatters mit dem Ausgang eines in der codeproduzierenden Reihe befindlichen Speicherelements (FF2), der zweite Eingang mit dem Ausgang eines weiteren in der codeproduzierenden Reihe befindlichen Speicherelements (FF5) und der Ausgang mit dem Eingang des in der codeproduzierenden

Reihe dem mit dem ersten Eingang des XOR-Gatters verbundenen Speicherelement nachfolgenden Speicherelements (FF3) verbunden ist.

34. Vorrichtung nach Anspruch 33, dadurch gekennzeichnet, dass in die den zweiten Eingang des wenigstens einen XOR-Gatters (XORp1) und den Ausgang des weiteren in der codeproduzierenden Reihe (21;24) befindlichen Speicherelements (FF5) verbindende Leitung ein UND-Gatter (UNDp1) derart geschalten ist, dass der Ausgang des UND-Gatters (UNDp1) mit dem zweiten Eingang des XOR-Gatters (XORp1), der erste Eingang des UND-Gatters (UNDp1) mit dem Ausgang des weiteren in der codeproduzierenden Reihe (21;24) befindlichen Speicherelements (FF5) und der zweite Eingang des UND-Gatters (UNDp1) mit dem Ausgang eines codeprogrammierenden Speicherelements (FFp2) verbunden ist und dass bevorzugt der Ausgang eines in der codeproduzierenden Reihe (21;24) befindlichen Speicherelements (FF9) mit dem Eingang eines Inverters (INV) und der Ausgang des Inverters (INV) mit dem Eingang eines anderen in der codeproduzierenden Reihe (21;24) angeordneten Speicherelements (FF1) verbunden ist, wobei als codeprogrammierendes Speicherelement ein Speicherelement eines weiteren rückgekoppelten Schieberegisters (22;25) verwendet wird.

35. Vorrichtung nach Anspruch 33 oder 34, dadurch gekennzeichnet, dass eine Mehrzahl von XOR-Gattern (XORp1,p2,p3,p4) vorgesehen ist, deren erster Eingang jeweils von einem Ausgang eines in der codeproduzierenden Reihe (21;24) befindlichen Speicherelements (FF1,2,3,4) gespeist wird und deren zweiter Eingang jeweils vom Ausgang eines weiteren in der codeproduzierenden Reihe (21;24) befindlichen Speicherelements (FF8,15,20,23) gespeist wird, welches eine Anzahl von Speicherelementen in Flussrichtung der Reihe (21;24) von dem

jeweils mit dem ersten Eingang verbundenen Speicherelement (FF1,2,3,4) entfernt ist, welche jeweils einer unterschiedlichen Primzahl entspricht, die größer als 1 und kein Teilbetrag der Gesamtzahl der in Reihe (21;24) geschalteten Speicherelemente (FF1,2,...n) ist.

36. Vorrichtung nach einem der Ansprüche 33 bis 35, dadurch gekennzeichnet, dass eine Mehrzahl von codeprogrammierenden, jeweils einem UND-Gatter (UNDp1,p2,p3,p4) und einem XOR-Gatter (XORp1,p2,p3,p4) zugeordneten Speicherelementen (FFp1,p2,p3,p4,...pn) vorgesehen und in einer zu einem Kreis geschlossenen Reihe (22;25) geschalten ist und wenigstens ein XOR-Gatter (XORpp1) angeordnet ist, dessen erster Eingang mit dem Ausgang eines in der codeprogrammierenden Reihe (22;25) befindlichen Speicherelements (FFp6), dessen zweiter Eingang mit dem Ausgang eines weiteren in der codeprogrammierenden Reihe (22;25) befindlichen Speicherelements (FFp5) und dessen Ausgang mit dem Eingang des in der codeprogrammierenden Reihe (22;25) dem mit dem ersten Eingang des XOR-Gatters (XORpp1) verbundenen Speicherelement (FFp6) nachfolgenden Speicherelements (FFp1) verbunden ist.

37. Vorrichtung nach einem der Ansprüche 33 bis 36, dadurch gekennzeichnet, dass in die den zweiten Eingang des wenigstens einen XOR-Gatters (XORpp1) und den Ausgang des weiteren in der codeprogrammierenden Reihe (22;25) befindlichen Speicherelements (FFp3) verbindende Leitung ein UND-Gatter (UNDpp1) derart geschalten ist, dass der Ausgang des UND-Gatters (UNDpp1) mit dem zweiten Eingang des XOR-Gatters (XORpp1), der erste Eingang des UND-Gatters (UNDpp1) mit dem Ausgang des weiteren in der codeprogrammierenden Reihe (22;25) befindlichen Speicherelements (FFp3) und der zweite Eingang des UND-Gatters (UNDpp1) mit dem Ausgang eines der

Programmierung der codeprogrammierenden Reihe (22;25) dienenden Speicherelements (FFpp5) verbunden ist.

38. Vorrichtung nach einem der Ansprüche 33 bis 37, dadurch gekennzeichnet, dass eine Mehrzahl von der Programmierung der codeprogrammierenden Reihe (22;25) dienenden, jeweils einem UND-Gatter (UNDpp1) und einem XOR-Gatter (XORpp1) zugeordneten Speicherelementen (FFpp1,pp2,pp3,pp4,...ppn) vorgesehen und in einer zu einem Kreis geschlossenen Reihe (23;26) geschalten ist und wenigstens ein XOR-Gatter (XORppp1) angeordnet ist, dessen erster Eingang mit dem Ausgang eines in der Reihe (23;26) befindlichen Speicherelements (FFpp1), dessen zweiter Eingang mit dem Ausgang eines weiteren in der Reihe (23;26) befindlichen Speicherelements (FFpp3) und dessen Ausgang mit dem Eingang des in der Reihe (23;26) dem mit dem ersten Eingang des XOR-Gatters (XORppp1) verbundenen Speicherelement (FFpp1) nachfolgenden Speicherelements (FFpp2) verbunden ist.

39. Datenbestand, insbesondere Datenbank (1), umfassend eine Daten enthaltende Datenstruktur (2) und eine zugehörige Indexdaten enthaltende Indexstruktur (3), wobei die Daten in der Datenstruktur (2) und die Indexdaten in der Indexstruktur (3) mittels einer Stromchiffrierung verschlüsselt gespeichert sind.

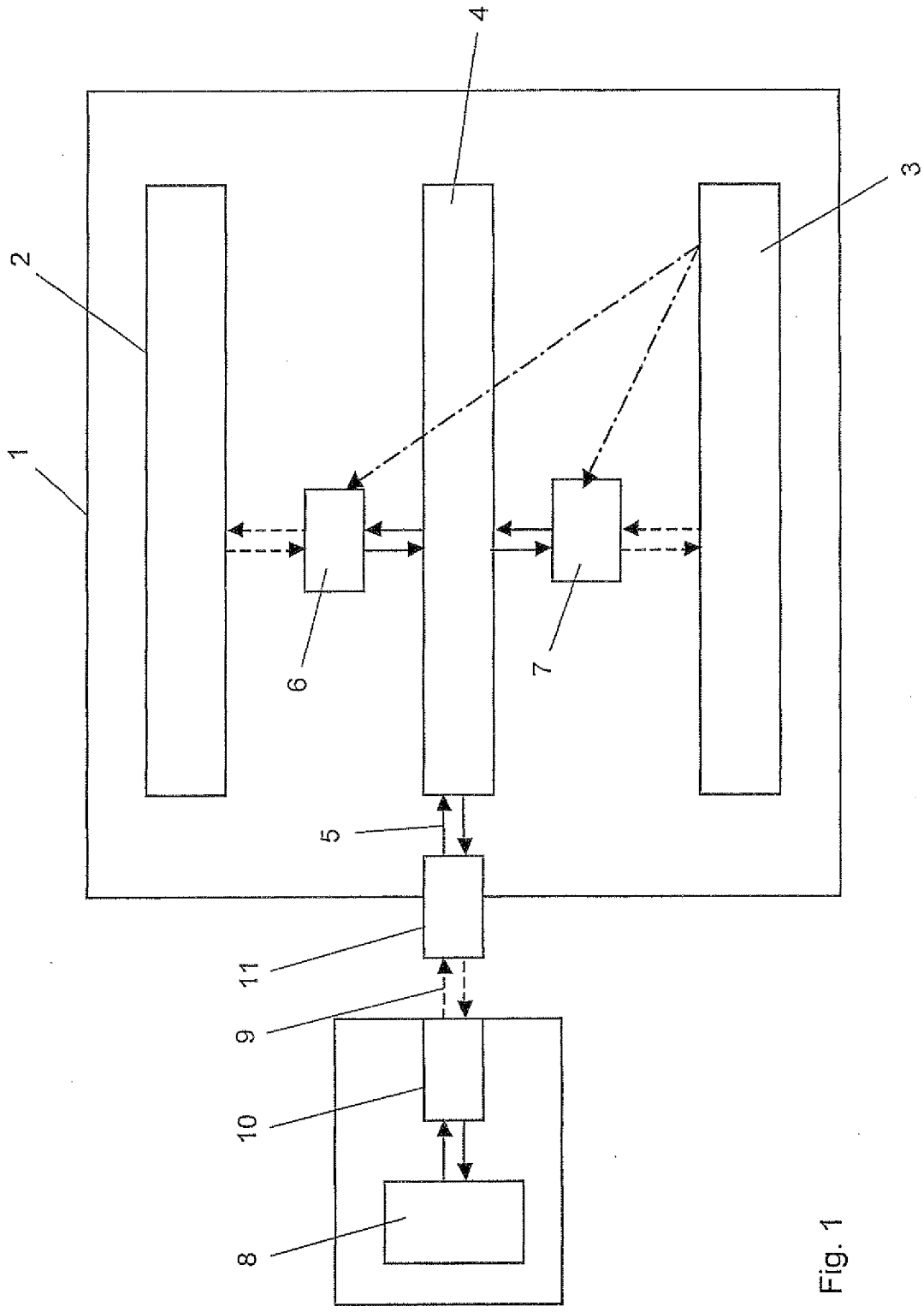


Fig. 1

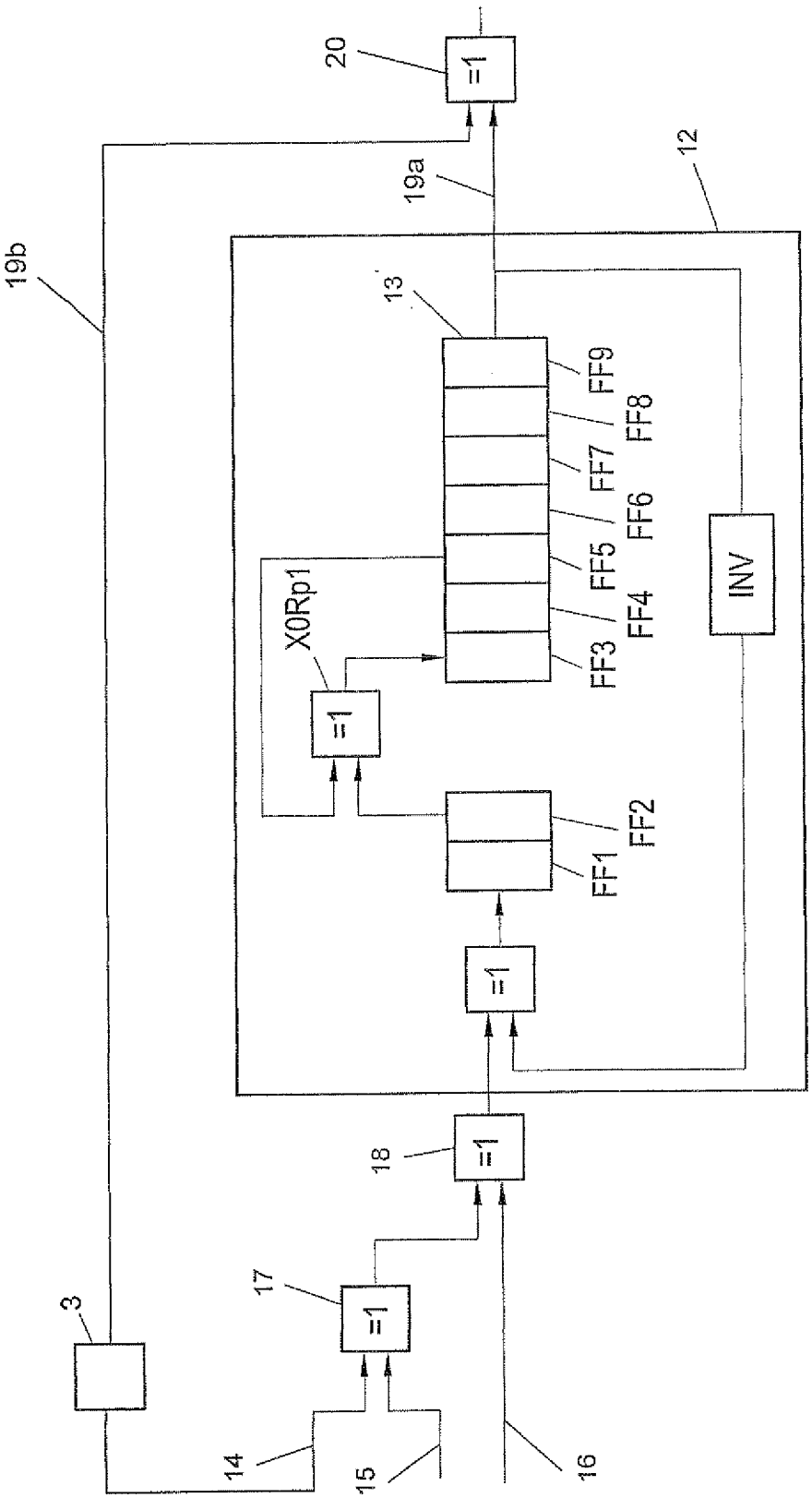

20
=1
19a
12
19b
13
FF9
FF8
FF7
FF6
FF5
FF4
FF3
FF2
FF1
INV
X0Rp1
=1
=1
=1
18
17
3
14
15
16

Fig. 2

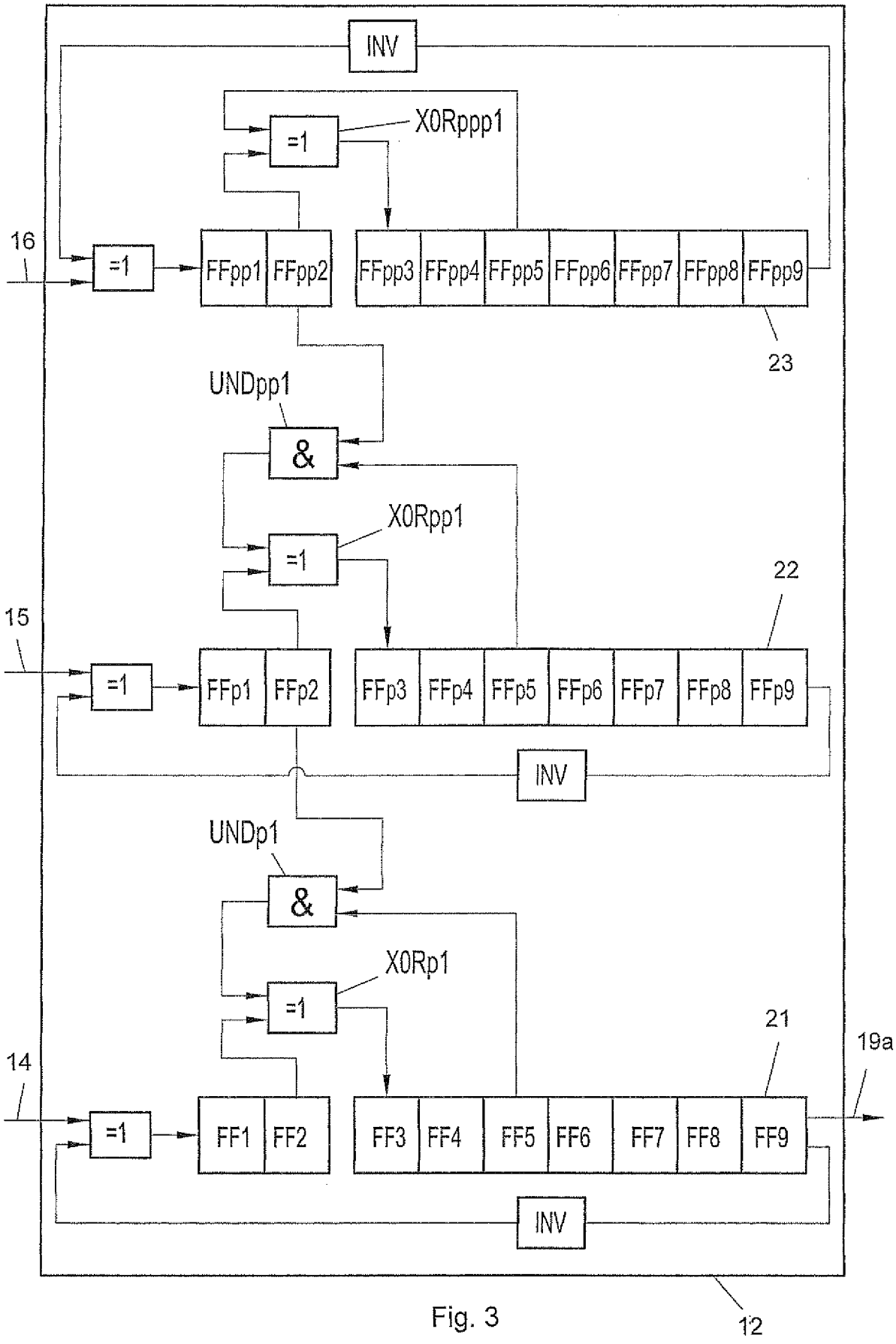

INV
=1
X0Rppp1
16
=1
FFpp1
FFpp2
FFpp3
FFpp4
FFpp5
FFpp6
FFpp7
FFpp8
FFpp9
23
UNDpp1
&
X0Rpp1
=1
22
15
=1
FFp1
FFp2
FFp3
FFp4
FFp5
FFp6
FFp7
FFp8
FFp9
INV
UNDp1
&
X0Rp1
=1
21
19a
14
=1
FF1
FF2
FF3
FF4
FF5
FF6
FF7
FF8
FF9
INV
12

Fig. 3

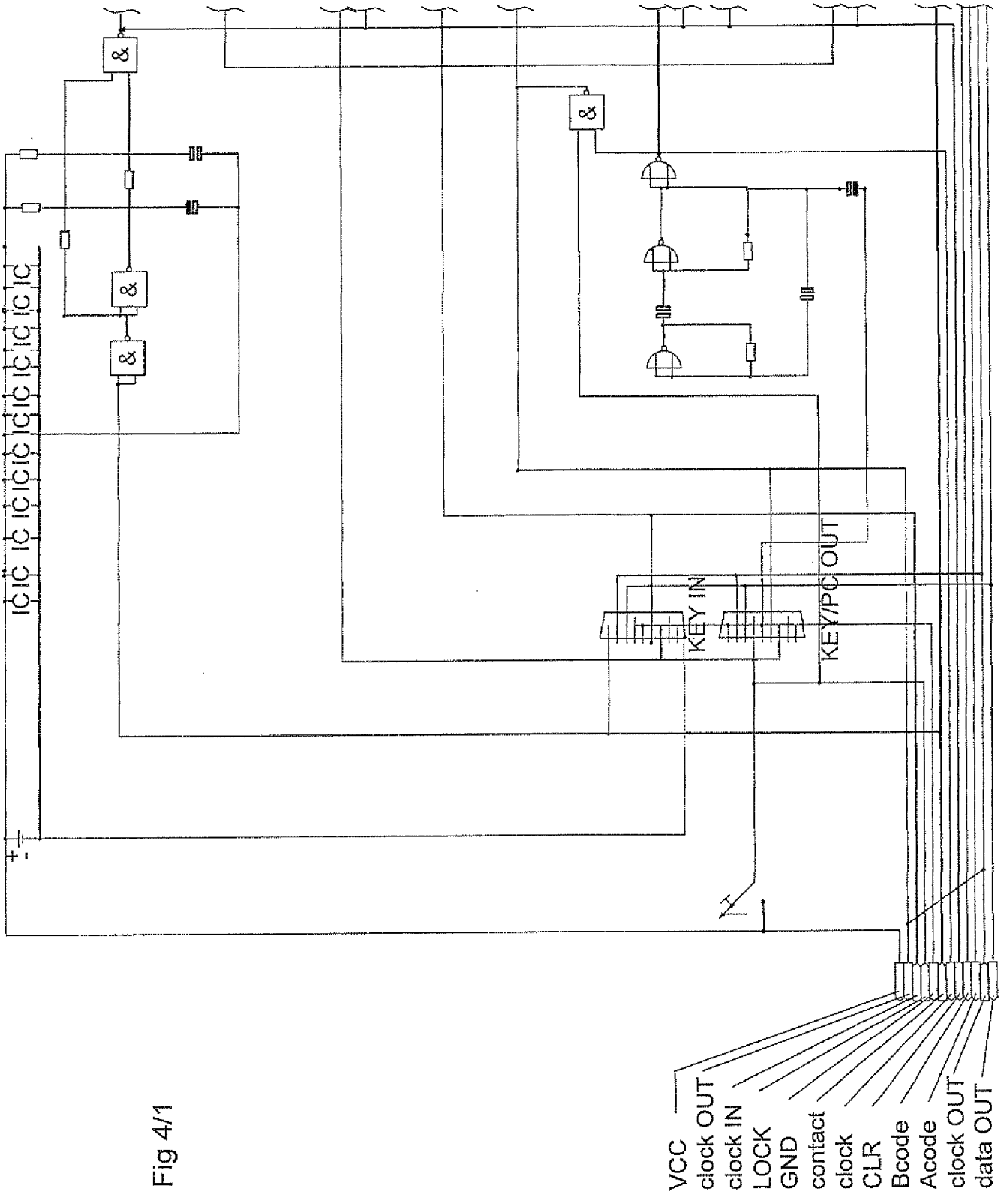
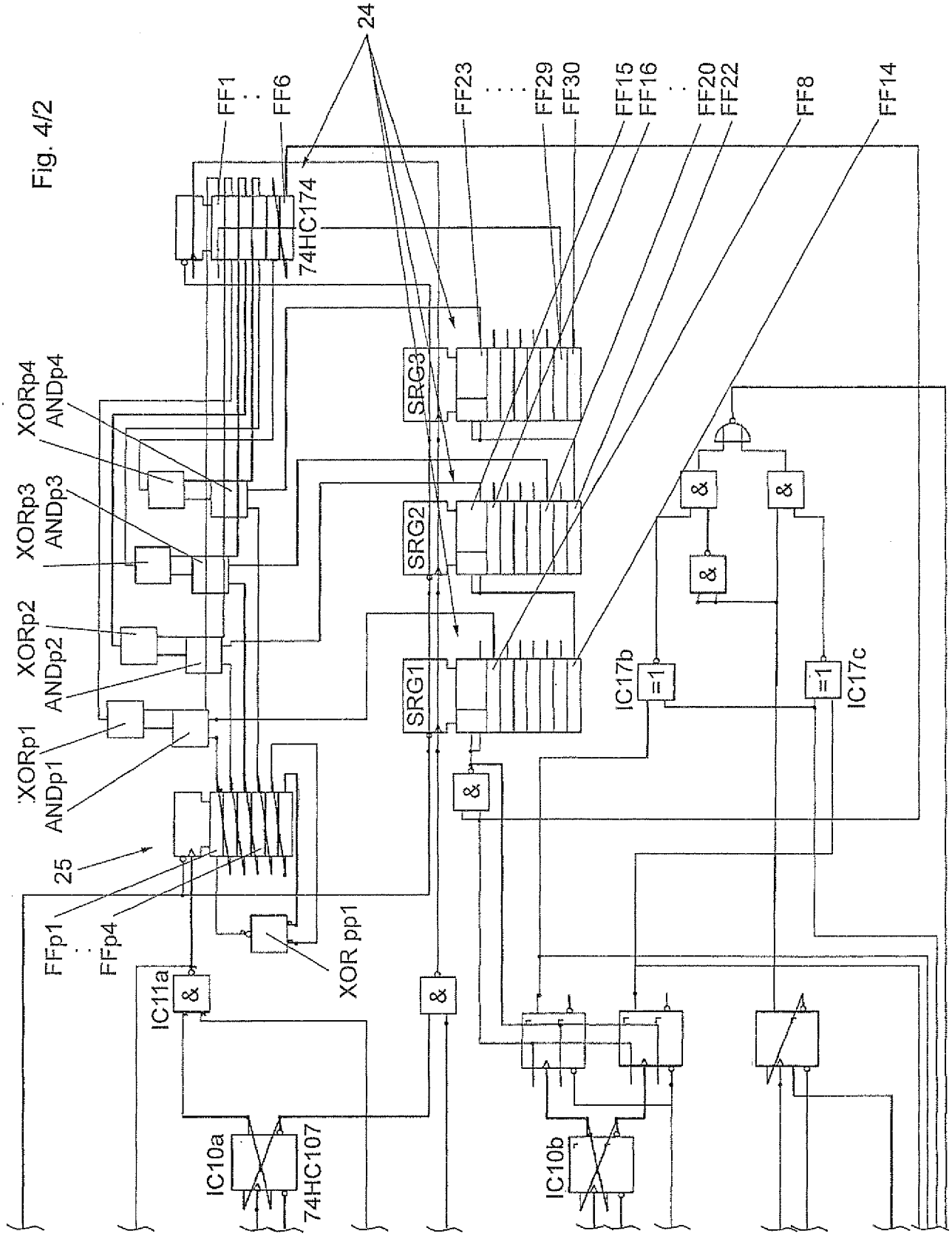

&
&
&
&
IC IC IC IC IC IC IC IC IC IC IC IC
KEY IN
KEY/PC OUT
VCC
clock OUT
clock IN
LOCK
GND
contact
clock
CLR
Bcode
Acode
clock OUT
data OUT

Fig 4/1

Fig. 4/2


24
FF1
FF6
FF23
FF29
FF30
FF15
FF16
FF20
FF22
FF8
FF14
74HC174
SRG3
SRG2
SRG1
XORp4
ANDp4
XORp3
ANDp3
XORp2
ANDp2
XORp1
ANDp1
25
FFp1
FFp4
XOR pp1
IC11a
IC10a
74HC107
IC10b
IC17b
IC17c
=1
&

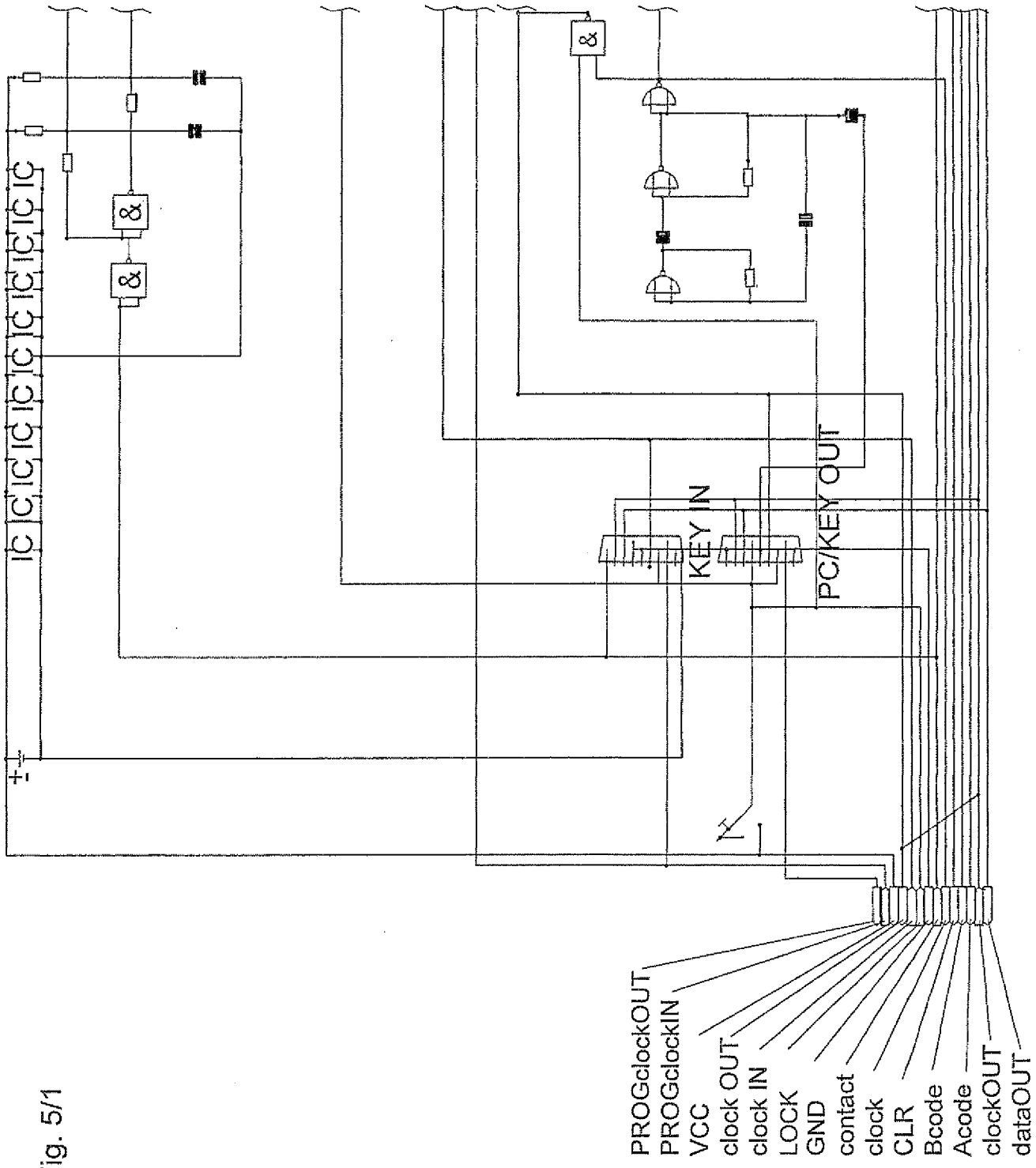
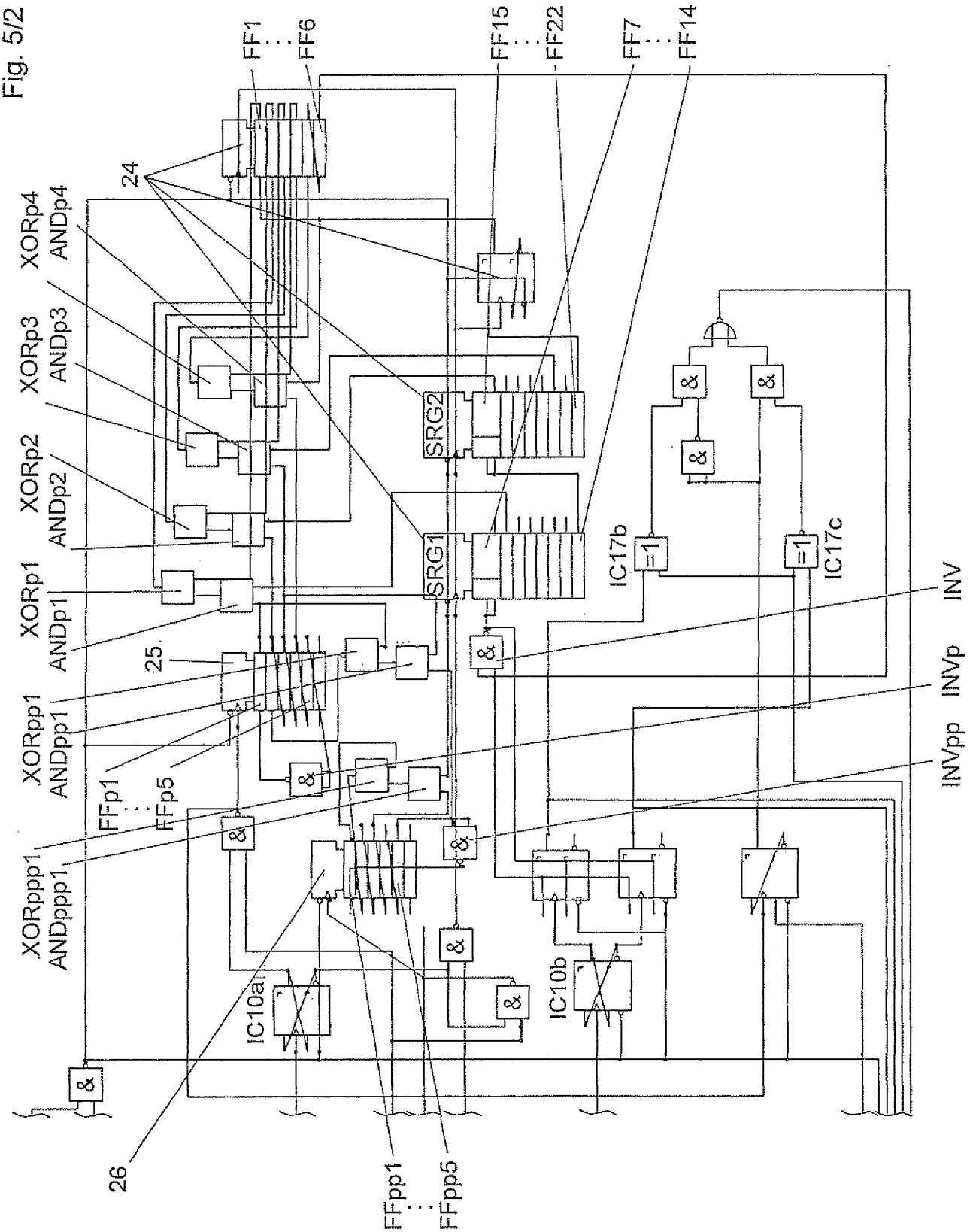

IC
IC
IC
IC
IC
IC
IC
IC
IC
IC
IC
IC
&
&
&
KEY IN
PC/KEY OUT
+
-
PROGclockOUT
PROGclockIN
VCC
clock OUT
clock IN
LOCK
GND
contact
clock
CLR
Bcode
Acode
clockOUT
dataOUT

Fig. 5/1

Fig. 5/2


FF1
FF6
FF15
FF22
FF7
FF14
24
XORp4
ANDp4
XORp3
ANDp3
XORp2
ANDp2
XORp1
ANDp1
XORpp1
ANDpp1
XORppp1
ANDppp1
SRG2
SRG1
IC17b
IC17c
=1
=1
INV
INVp
INVpp
25
FFp1
FFp5
IC10a
IC10b
26
FFpp1
FFpp5
&