(19) österreichisches patentamt

(10) **AT 510 730 B1 2013-06-15**

(12) **Patentschrift**

(21) Anmeldenummer: A 2007/2010
(22) Anmeldetag: 02.12.2010
(45) Veröffentlicht am: 15.06.2013

(51) Int. Cl. : ***H04L 9/26*** (2006.01)

(56) Entgegenhaltungen:
EP 0615361 A1 US 5859912 A
US 2003091185 A1
WO 2003075507 A1

(73) Patentinhaber:
CORDES RENE-MICHAEL MAG.
2323 MANNSWÖRTH (AT)
SCHOBESBERGER ERNESTO DR.
1040 WIEN (AT)

(72) Erfinder:
CORDES RENE-MICHAEL MAG.
MANNSWÖRTH (AT)
SCHOBESBERGER ERNESTO DR.
WIEN (AT)

(54) VERFAHREN UND VORRICHTUNG ZUR DURCHFÜHRUNG EINER SYMMETRISCHEN STROMVERSCHLÜSSELUNG VON DATEN

(57) Bei einem Verfahren zur Durchführung einer symmetrischen Stromverschlüsselung von Daten unter Verwendung eines Schlüsselstroms und zur Übertragung der verschlüsselten Daten, wobei die Generierung des Schlüsselstroms unter Verwendung wenigstens eines rückgekoppelten Schieberegisters erfolgt, das zu seiner Initialisierung mit einer definierten Bitfolge gefüllt wird, werden die zu verschlüsselnden Daten in Datenpakete aufgeteilt werden, wobei jedes Datenpaket gesondert verschlüsselt wird. Das bzw. die rückgekoppelte (n) Schieberegister wird bzw. werden für die Verschlüsselung jedes Datenpakets neu initialisiert, wobei zur Initialisierung des bzw. der rückgekoppelten Schieberegister jeweils wenigstens eine erste Bitfolge und eine zweite Bitfolge verwendet wird, wobei die erste Bitfolge dem jeweils verschlüsselten Datenpaket im Klartext oder in codierter Form hinzugefügt wird und die zweite Bitfolge einen geheimen Schlüssel darstellt, die den verschlüsselten Datenpaketen nicht hinzugefügt wird. Die verschlüsselten Datenpakete werden samt der jeweiligen hinzugefügten Bitfolge und ggf. Kopfdaten paketvermittelt übertragen.

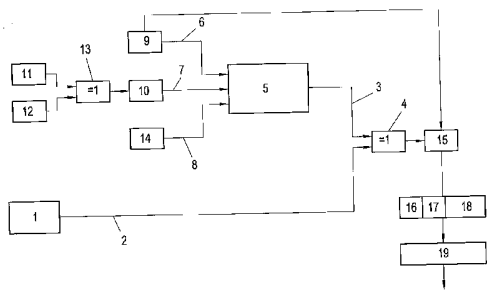


Fig. 1

AT 510 730 B1 2013-06-15

Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zur Durchführung einer symmetrischen Stromverschlüsselung von Daten unter Verwendung eines Schlüsselstroms und zur Übertragung der verschlüsselten Daten, wobei die Generierung des Schlüsselstroms unter Verwendung wenigstens eines rückgekoppelten Schieberegisters erfolgt, das zu seiner Initialisierung mit einer definierten Bitfolge gefüllt wird.

[0002] Die Erfindung betrifft weiters ein entsprechendes Verfahren zum Entschlüsseln von mittels einer symmetrischen Stromverschlüsselung verschlüsselten Daten.

[0003] Die Erfindung betrifft weiters eine Vorrichtung zum Verschlüsseln von Daten mit Hilfe einer symmetrischen Stromverschlüsselung unter Verwendung eines Schlüsselstroms, wobei zur Generierung des Schlüsselstroms wenigstens ein rückgekoppeltes Schieberegister vorgesehen ist, das zu seiner Initialisierung jeweils mit einer definierten Bitfolge gefüllt wird. Die Erfindung betrifft weiters eine Vorrichtung zum Entschlüsseln von mittels einer symmetrischen Stromverschlüsselung verschlüsselten Daten.

[0004] 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.

[0005] Zur Erzeugung des Schlüsselstroms wird in der Regel wenigstens ein rückgekoppeltes Schieberegister verwendet. 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.

[0006] 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.

[0007] 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.

[0008] 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 wesentliche Verbesserung wird in dieser Hinsicht durch den aus der WO 03/075507 A1 bekannten Codegenerator erzielt.

[0009] Ein weiterer Nachteil bei der herkömmlichen Stromverschlüsselung von Daten ist der Umstand, dass sie bei der paketvermittelten Datenübertragung (z.B. im Internet über das IP-Protokoll) nur unter Verringerung der Verschlüsselungssicherheit verwendet werden kann. Bei der paketvermittelten Datenübertragung muss jedes Datenpaket gesondert verschlüsselt werden und der für die Verschlüsselung jedes Datenpakets verwendete Schlüssel muss beim Empfänger zum Zwecke der Entschlüsselung bekannt sein, um eine Entschlüsselung auch dann zu ermöglichen, wenn einzelne Datenpakete verloren gehen, Pakete doppelt beim Empfänger ankommen, Pakete verschiedene Wege nehmen oder Pakete fragmentiert beim Empfänger ankommen. Die einfachste Möglichkeit, auch unter den Bedingungen dieser Fehlfunktionen eine eindeutige Zuordnung von Daten mit dem Schlüssel bzw. Schlüsselstrom zu gewährleisten, ist es, für jedes Datenpaket denselben Schlüsselstrom zu verwenden. Dies erleichtert es aber, die Verschlüsselung zu brechen.

[0010] Die vorliegende Erfindung zielt daher darauf ab, ein Verfahren und eine Vorrichtung zum Ver- und/oder Entschlüsseln von Daten unter Verwendung einer Stromver- bzw. -entschlüsselung zu schaffen, wobei die verschlüsselten Daten in Paketen beliebiger Größe vorliegen sollen, so dass diese zur simultanen Übermittlung von binären Datenströmen hoher Frequenz über lange Zeiträume in paketorganisierten Datennetzen geeignet sind. Die Verschlüsselung soll so sicher wie möglich sein, wobei ein Brechen der Verschlüsselung so gut wie verunmöglicht werden soll.

[0011] Zur Lösung dieser Aufgabe ist gemäß einem ersten Aspekt der Erfindung das Verschlüsselungsverfahren der eingangs genannten Art dahingehend weitergebildet, dass die zu verschlüsselnden Daten in Datenpakete aufgeteilt werden, dass jedes Datenpaket gesondert verschlüsselt wird, wobei das bzw. die rückgekoppelte(n) Schieberegister für die Verschlüsselung jedes Datenpakets neu initialisiert wird bzw. werden, wobei zur Initialisierung des bzw. der rückgekoppelten Schieberegister jeweils wenigstens eine erste Bitfolge und eine zweite Bitfolge verwendet wird, wobei die erste Bitfolge dem jeweils verschlüsselten Datenpaket im Klartext oder in codierter Form hinzugefügt wird und die zweite Bitfolge einen geheimen Schlüssel darstellt, die den verschlüsselten Datenpaketen nicht hinzugefügt wird, und dass die verschlüsselten Datenpakete samt der jeweiligen hinzugefügten Bitfolge und ggf. Kopfdaten paketvermittelt übertragen werden.

[0012] Zum Entschlüsseln der Datenpakete ist gemäß einem zweiten Aspekt der Erfindung vorgesehen, dass die zu entschlüsselnden Daten als Datenpakete empfangen werden, dass jedes empfangene Datenpaket gesondert entschlüsselt wird, wobei das bzw. die rückgekoppelte(n) Schieberegister für die Entschlüsselung jedes Datenpakets neu initialisiert wird bzw. werden, wobei zur Initialisierung des bzw. der rückgekoppelten Schieberegister jeweils wenigstens eine erste Bitfolge und eine zweite Bitfolge verwendet wird, wobei die erste Bitfolge aus dem jeweils zu entschlüsselnden Datenpaket im Klartext oder in codierter Form ausgelesen wird und die zweite Bitfolge einen geheimen Schlüssel darstellt, die aus den zu entschlüsselnden Datenpaketen nicht ausgelesen werden kann.

[0013] Erfindungsgemäß wird somit sämtliche für die Entschlüsselung jedes einzelnen Datenpakets erforderliche Information mit Ausnahme des geheimen Schlüssels im jeweiligen Paket mitgeführt, sodass die Kommunikationspartner lediglich den geheimen Schlüssel bzw. die für die Generierung des geheimen Schlüssels erforderlichen Informationen vor der Datenübermittlung austauschen müssen. Dadurch, dass erfindungsgemäß jedes Datenpaket die erste Bitfolge im Klartext oder in codierter Form enthält, die zur Verschlüsselung des jeweiligen Datenpakets verwendet wurde, kann diese erste Bitfolge vom Empfänger aus dem jeweiligen Datenpaket ausgelesen und zum Entschlüsseln herangezogen werden. Dies ermöglicht es, jedes Datenpa-

ket mit einem anderen Schlüssel zu verschlüsseln, sodass ein Brechen der Verschlüsselung erschwert wird. Wenn, wie dies einer bevorzugten Verfahrensweise entspricht, als erste Bitfolge eine für das zu verschlüsselnde Datenpaket eindeutige Bitfolge gewählt wird, die dem jeweils verschlüsselten Datenpaket als Paketkennung im Klartext oder in codierter Form hinzugefügt wird, wird sichergestellt, dass zwei Datenpakete mit dem gleichen Klartextinhalt nicht auch identisch verschlüsselt werden, die verschlüsselten Datenpakete sich also voneinander unterscheiden. Rückschlüsse auf die übertragenen Zeichen durch eine statistische Auswertung der Datenpakete werden dadurch erschwert.

[0014] Die zweite Bitfolge, d.h. der geheime Schlüssel, wird bevorzugt aus einer eindeutigen Kennung des Senders und einer eindeutigen Kennung des Empfängers generiert. Als eindeutige Kennung kann hierbei beispielsweise eine Hardwarekennung des Senders bzw. des Empfängers herangezogen werden, insbesondere eine herstellseitig eingeprägte Chipnummer oder dgl. Die Generierung der zweiten Bitfolge erfolgt bevorzugt durch Verknüpfung der eindeutigen Kennung des Senders und der eindeutigen Kennung des Empfängers mit Hilfe einer XOR-Funktion. Hierzu ist es erforderlich, dass die Sender und Empfänger vor der Datenübertragung ihre Kennungen austauschen.

[0015] Wie erwähnt werden die erste und die zweite Bitfolge im Rahmen der Ver- bzw. Entschlüsselung dazu verwendet, das oder die rückgekoppelte(n) Schieberegister zu initialisieren. 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. Diese Vorgehensweise erschwert es, auf Grundlage der im Klartext mitübermittelten ersten Bitfolge die Struktur des Schlüsselstromgenerators und/oder den geheimen Schlüssel zu ermitteln.

[0016] 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 wird dabei mit Vorteil aus einer jeweils aktuellen Datums- und/oder Zeitangabe generiert. Die dritte Bitfolge wird bevorzugt zur Initialisierung einem dritten rückgekoppelten Schieberegister zugeführt.

[0017] 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.

[0018] 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.

[0019] 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 31 bis 36 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.

[0020] Gemäß einem weiteren Aspekt der vorliegenden Erfindung werden eine Verschlüsselungs- und eine Entschlüsselungsvorrichtung vorgeschlagen.

[0021] Die erfindungsgemäße Vorrichtung zum Verschlüsseln von Daten mit Hilfe einer symmetrischen Stromverschlüsselung unter Verwendung eines Schlüsselstroms, wobei zur Generierung des Schlüsselstroms wenigstens ein rückgekoppeltes Schieberegister vorgesehen ist, das zu seiner Initialisierung jeweils mit einer definierten Bitfolge gefüllt wird, ist dadurch gekennzeichnet, dass die Daten in Datenpakete aufgeteilt vorliegen, dass Mittel zum Generieren und/oder Speichern wenigstens einer ersten Bitfolge und einer zweiten Bitfolge vorgesehen sind, die mit dem bzw. den Schieberegister(n) derart zusammenwirken, dass wenigstens die erste Bitfolge und die zweite Bitfolge zur Initialisierung des bzw. der rückgekoppelten Schieberegister verwendet werden, wobei das bzw. die rückgekoppelte(n) Schieberegister für die Verschlüsselung jedes Datenpakets neu initialisiert wird bzw. werden, dass Datenpaketverarbeitungsmittel vorgesehen sind, mit denen die Mittel zum Generieren bzw. Speichern der ersten und der zweiten Bitfolge derart zusammenwirken, dass die erste Bitfolge dem jeweils verschlüsselten Datenpaket im Klartext oder in codierter Form hinzugefügt wird und die zweite Bitfolge einen geheimen Schlüssel darstellt, die den verschlüsselten Datenpaketen nicht hinzugefügt wird, und dass Datenübertragungsmittel zum paketvermittelten Versenden der verschlüsselten Datenpakete samt der jeweiligen hinzugefügten Bitfolge und ggf. Kopfdaten vorgesehen sind.

[0022] Die erfindungsgemäße Vorrichtung zum Entschlüsseln von mittels einer symmetrischen Stromverschlüsselung verschlüsselten Daten unter Verwendung eines Schlüsselstroms, wobei zur Generierung des Schlüsselstroms wenigstens ein rückgekoppeltes Schieberegister vorgesehen ist, das zu seiner Initialisierung jeweils mit einer definierten Bitfolge gefüllt wird, ist dadurch gekennzeichnet, dass die verschlüsselten Daten in Datenpakete aufgeteilt vorliegen, dass Mittel zum Auslesen einer ersten Bitfolge im Klartext oder in codierter Form aus den Datenpaketen und Mittel zum Generieren und/oder Speichern wenigstens einer zweiten Bitfolge vorgesehen sind, die mit dem bzw. den Schieberegister (n) derart zusammenwirken, dass wenigstens die erste Bitfolge und die zweite Bitfolge zur Initialisierung des bzw. der rückgekoppelten Schieberegister verwendet werden, wobei das bzw. die rückgekoppelte(n) Schieberegister für die Entschlüsselung jedes Datenpakets neu initialisiert wird bzw. werden, wobei die zweite Bitfolge einen geheimen Schlüssel darstellt, die aus den verschlüsselten Datenpaketen nicht ausgelesen werden kann.

[0023] Bevorzugte Weiterbildungen ergeben sich aus den Unteransprüchen.

[0024] Die Erfindung wird in der Folge anhand von in der Zeichnung schematisch dargestellten Ausführungsbeispielen näher erläutert. In dieser zeigen

[0025] Fig.1 eine erfindungsgemäße Verschlüsselungsvorrichtung,

[0026] Fig.2 eine erfindungsgemäße Entschlüsselungsvorrichtung,

[0027] Fig.3, Fig.4, Fig.5 und Fig.6 verschiedene Ausbildungen eines in der Vorrichtung verwendeten Schlüsselstromgenerators.

[0028] In Fig. 1 ist ein zu verschlüsselndes Datenpaket mit 1 bezeichnet, wobei das Datenpaket 1 eine Vielzahl von Bits im Klartext umfasst. Die Verschlüsselung erfolgt grundsätzlich derart, dass die Bits des Bitstroms 2 des Klartextes mit den Bits eines Schlüsselstroms 3 einzeln mit Hilfe eines XOR-Gatters 4 verknüpft werden. Der Erzeugung des Schlüsselstroms 3 dient ein Codegenerator 5, der an Hand der Fig. 3 bis 6 noch näher beschrieben werden wird. Der Codegenerator 5 erzeugt den Schlüsselstrom 3 auf Grundlage einer Mehrzahl von Bitfolgen 6, 7 und 8, die dem Codegenerator 5 als Schlüssel zugeführt werden. Eine erste Bitfolge 6 ist in einem Speicher 9 gespeichert und stellt eine einzigartige Kennung des zu verschlüsselnden Datenpakets 1 dar. Die Einzigartigkeit muss hierbei zumindest innerhalb der Gesamtzahl der zusammenhängend zu übermittelten Datenpakete gegeben sein. Die Länge der ersten Bitfolge

beträgt somit mindestens log(N;2) Bit (N = Gesamtzahl der übermittelten Pakete). Die zweite Bitfolge 7 ist in einem Speicher 10 gespeichert und wird aus einer eindeutigen Kennung 11 des Senders und einer eindeutigen Kennung 12 des Empfängers generiert. Die Erzeugung der zweiten Bitfolge 7 erfolgt dabei dadurch, dass die Bits der eindeutigen Kennung 11 und die Bits der eindeutigen Kennung 12 mit Hilfe eines XOR-Gatters 13 miteinander verknüpft werden. Auf Grund der Verwendung der zweiten Bitfolge 7 als Schlüssel für die Erzeugung des Schlüsselstroms 3 wird sichergestellt, dass nur der Empfänger, dem ebenfalls die eindeutigen Kennungen 11 und 12 bekannt sein müssen, die verschlüsselten Datenpakete entschlüsseln kann. Die dritte Bitfolge 8 ist in einem Speicher 14 gespeichert bzw. wird dort generiert, und zwar auf Grundlage einer aktuellen Datums- oder Zeitangabe. Beispielsweise entspricht die Bitfolge 8 dem aktuellen Datum. Dies führt dazu, dass der Schlüsselstrom 3 jeden Tag eine gänzlich andere Struktur aufweist, sodass ein Brechen der Verschlüsselung erschwert wird.

[0029] Die verschlüsselten Daten des Datenpakets werden nun Datenpaketverarbeitungsmitteln 15 zugeführt, mit denen der Speicher 9 für die erste Bitfolge 6 derart zusammenwirkt, dass die erste Bitfolge 6 dem verschlüsselten Datenpaket im Klartext hinzugefügt wird. Die zweite Bitfolge 7 und die dritte Bitfolge 8 hingegen werden dem verschlüsselten Datenpaket nicht hinzugefügt, sondern sind beim Empfänger ohnehin bekannt. Die Datenpaketverarbeitungsmittel 15 sorgen weiters dafür, dass das verschlüsselte Datenpaket mit den üblichen Kopfdaten versehen wird, die für die paketvermittelte Übertragung in einem Computernetzwerk erforderlich sind. Das für die Versendung vorbereitete Datenpaket besteht somit aus Kopfdaten 16, der ersten Bitfolge als Paketkennung 17 und den verschlüsselten Nutzdaten 18. Die Datenübertragungsmittel zum paketvermittelten Versenden des Datenpakets sind mit 19 bezeichnet.

[0030] Die in Fig. 2 dargestellte Vorrichtung zum Entschlüsseln der verschlüsselten Datenpakete ist im Wesentlichen analog aufgebaut. Das die Kopfdaten 16, die ersten Bitfolge als Paketkennung 17 und die verschlüsselten Nutzdaten 18 enthaltende Paket wird beim Eintreffen Auslesemitteln 20 zugeführt, in denen die zweite Bitfolge 17 ausgelesen und einem Speicher 21 zugeführt wird. Die verschlüsselten Nutzdaten 18 werden in der Folge einem XOR-Gatter 22 zugeführt, in dem die Bits des verschlüsselten Bitstroms 23 und die Bits des Schlüsselstroms 3 miteinander verknüpft werden, um auf diese Weise das entschlüsselte Datenpaket 1 zu erhalten.

[0031] Der Schlüsselstrom 3, der für die Entschlüsselung eines bestimmten Datenpakets verwendet wird, muss der gleiche sein wie der Schlüsselstrom, der für die Verschlüsselung dieses Datenpakets verwendet wurde. Zu diesem Zweck werden dieselben Bitfolgen 6, 7 und 8 dem Generator 5 als Schlüssel zugeführt und der für die Entschlüsselung verwendete Generator 5 ist baugleich mit dem für die Verschlüsselung verwendeten Generator 5. Der Speicher für die zweite Bitfolge 7 ist mit 24 bezeichnet. Dem Speicher 24 sind die über das XOR-Gatter 25 miteinander verknüpften Sender- und Empfängerkennungen 11 und 12 zugeführt. Die dritte Bitfolge 8 ist im Speicher 26 gespeichert bzw. wird dort generiert.

[0032] Fig. 3 zeigt eine Prinzipschaltung eines Schlüsselstromgenerators 5 mit einem Schieberegister 27, 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 der XOR-Gatters XORp1 mit dem Eingang des in Flussrichtung dem 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 27 mit einer Bitfolge befüllt, erhält man mit dieser Schaltung eine Codesequenz. Wenn, wie dies bei der Ausbildung gemäß Fig. 3 der Fall ist, nur ein einziges Schieberegister zum Einsatz gelangt, werden die Bitfolgen 6, 7 und 8 dem Schieberegister 27 derart zugeführt, dass zunächst die Bitfolgen 6 und 7 mit Hilfe eines XOR-Gatters 28 miteinander verknüpft werden und

dann die verknüpfte Bitfolge mit der Bitfolge 8 mit Hilfe des XOR-Gatters 29 verknüpft wird. Dabei ist es bevorzugt, dass die aus den Bitfolgen 6, 7 und 8 generierte, dem Schieberegister 27 zugeführte Bitfolge nicht länger ist als dies der Anzahl der Speicherelemente im Schieberegister 27 entspricht, da die Bitfolge andernfalls von der über den Inverter INV aus dem Speicherelemente FF9 kommenden Bitfolge überlagert würde.

[0033] Bei der abgewandelten Ausbildung gemäß Fig. 4 gelangen insgesamt drei Schieberegister 30, 31 und 32 zum Einsatz. Die Speicherelemente der einzelnen Schieberegister sind in diesem Beispiel jeweils auf gleiche Weise rekursiv verschaltet wie in Fig. 3. Die Schieberegister sind weiters derart miteinander verschaltet, dass in Abhängigkeit vom Zustand des zweiten Schieberegisters 31 die Funktion des XOR-Gatters XORp1 der rekursiven Verschaltung des ersten Schieberegisters 30 an- und abgeschaltet wird. Die Funktion des XOR-Gatters XORpp1 der rekursiven Verschaltung des zweiten Schieberegisters 31 wird wiederum in Abhängigkeit vom Zustand des dritten Schieberegisters 32 an- und abgeschaltet. Zu diesem Zweck ist der Ausgang des Flip-Flops FFp2 bzw. FFpp2 des einen Schiebregisters 31 bzw. 32 mit dem Eingang eines UND-Gatters UNDp1 bzw. UNDpp1 verbunden, das in die jeweilige rekursive Funktion XORp1 bzw. XORpp1 der Schieberegister 30 bzw. 31 eingefügt ist.

[0034] Es entsteht somit ein Codegenerator 5 mit drei Ebenen, wobei die Codegenerierung auf jeder Ebene durch Initialisieren des jeweiligen Schieberegisters 30, 31 und 32 mit der Bitfolge 6, 7 und 8 beeinflusst wird. Die Initialisierung kann dabei bevorzugt so erfolgen, dass dem Schieberegister 30 der ersten Ebene die erste Bitfolge 6, dem Schieberegister 31 der zweiten Ebene die zweite Bitfolge 7 und dem Schieberegister 32 der dritten Ebene die dritte Bitfolge 8 zugeführt wird, wobei die Bitfolgen 6, 7 und 8 bevorzugt so definiert sind wie in den Fig. 1 und 2 beschrieben.

[0035] Bei der Ausbildung gemäß Fig. 5 ist die in Fig. 4 gezeigte Struktur noch komplexer ausgestaltet und es sind insbesondere längere codeproduzierende Reihen und eine Mehrzahl von rekursiven Verschaltungen vorgesehen. Dabei ist eine Anzahl ununterbrochen in Reihe geschalteter Speicherelemente in Form von Schieberegister SRG1, SRG2,... verwirklicht, die funktionell gesehen gemeinsam ein Schieberegister 33 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)

[0036] 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)

[0037] 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.

[0038] Wenn man nun bei entsprechender Länge des Schieberegisters 33 an mehreren Stellen dieses Schieberegisters 33 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).

[0039] 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.

[0040] 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 34 zusammengeschaltet sein. In weiterer Folge können die codeprogrammierenden Speicherelemente FFp1,p2,p3,p4 Schieberegisters 34 selbst wiederum mit Hilfe eines XOR-Gatters XORpp1 rekursiv verschaltet werden.

[0041] 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)

[0042] 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)

[0043] Beispiel:

[0044] 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).

[0045] Daraus folgt:

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

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

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

[0046] Mit anderen Worten man muss 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.

[0047] Indem man die codeprogrammierenden Speicherelemente (FFp1,p2,p3,p4,p5,p6) des Schieberegisters 34 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 bestimmte Zeitspanne, in der die codeprogrammierenden Speicherelemente mit einem Programmtakt versorgt werden.

[0048] 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 codeprogammierende XOR-Gatter XORpp1 selbst wiederum unter Zwischenschaltung eines UND-Gatters UNDpp1 mit einer Speicherelementen-Reihe RRR verbunden und somit programmierbar gemacht wird, wobei wiederum ein XOR-Gatter XORppp1 zur rekursiven Verschaltung des Schieberegisters 37 verwendet wird (Fig.6).

[0049] Ausgehend von obigen 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 zur Durchführung einer symmetrischen Stromverschlüsselung von Daten unter Verwendung eines Schlüsselstroms und zur Übertragung der verschlüsselten Daten, wobei die Generierung des Schlüsselstroms unter Verwendung wenigstens eines ersten rückgekoppelten Schieberegisters und eines zweiten rückgekoppelten Schieberegisters erfolgt, die zu ihrer Initialisierung jeweils mit einer definierten Bitfolge gefüllt werden, **dadurch gekennzeichnet**, dass die zu verschlüsselnden Daten in Datenpakete aufgeteilt werden, dass jedes Datenpaket gesondert verschlüsselt wird, wobei die rückgekoppelten Schieberegister für die Verschlüsselung jedes Datenpakets neu initialisiert werden, wobei das erste rückgekoppelte Schieberegister zu seiner Initialisierung mit der ersten Bitfolge gefüllt wird und das zweite rückgekoppelte Schieberegister zu seiner Initialisierung mit der zweiten Bitfolge gefüllt wird, wobei die erste Bitfolge dem jeweils verschlüsselten Datenpaket im Klartext oder in codierter Form hinzugefügt wird und die zweite Bitfolge einen geheimen Schlüssel darstellt, die den verschlüsselten Datenpaketen nicht hinzugefügt wird, und dass die verschlüsselten Datenpakete samt der jeweiligen hinzugefügten Bitfolge und ggf. Kopfdaten paketvermittelt übertragen werden.

2. Verfahren zum Entschlüsseln von mittels einer symmetrischen Stromverschlüsselung verschlüsselten Daten unter Verwendung eines Schlüsselstroms, wobei die Generierung des Schlüsselstroms unter Verwendung wenigstens eines ersten rückgekoppelten Schieberegisters und eines zweiten rückgekoppelten Schieberegisters erfolgt, die zu ihren Initialisierung jeweils mit einer definierten Bitfolge gefüllt werden, **dadurch gekennzeichnet**, dass die zu entschlüsselnden Daten als Datenpakete empfangen werden, dass jedes empfangene Datenpaket gesondert entschlüsselt wird, wobei die rückgekoppelten Schieberegister für die Entschlüsselung jedes Datenpakets neu initialisiert werden, wobei das erste rückgekoppelte Schieberegister zu seiner Initialisierung mit der ersten Bitfolge gefüllt wird und das zweite rückgekoppelte Schieberegister zu seiner Initialisierung mit der zweiten Bitfolge gefüllt wird, wobei die erste Bitfolge aus dem jeweils zu entschlüsselnden Datenpaket im Klartext oder in codierter Form ausgelesen wird und die zweite Bitfolge einen geheimen Schlüssel darstellt, die aus den zu entschlüsselnden Datenpaketen nicht ausgelesen werden kann.

3. Verfahren nach Anspruch 1 oder 2, **dadurch gekennzeichnet**, dass als erste Bitfolge eine für das zu verschlüsselnde Datenpaket eindeutige Bitfolge gewählt wird, die dem jeweils verschlüsselten Datenpaket als Paketkennung im Klartext oder in codierter Form hinzugefügt wird.

4. Verfahren nach einem der Ansprüche 1 bis 3, **dadurch gekennzeichnet**, dass die zweite Bitfolge aus einer eindeutigen Kennung des Senders und einer eindeutigen Kennung des Empfängers generiert wird.

5. Verfahren nach Anspruch 4, **dadurch gekennzeichnet**, dass die Generierung der zweiten Bitfolge durch Verknüpfung der eindeutigen Kennung des Senders und der eindeutigen Kennung des Empfängers mit Hilfe einer XOR-Funktion erfolgt.

6. Verfahren nach einem der Ansprüche 1 bis 5, **dadurch gekennzeichnet**, dass zur Initialisierung des bzw. der rückgekoppelten Schieberegister weiters eine dritte Bitfolge verwendet wird.

7. Verfahren nach Anspruch 6, **dadurch gekennzeichnet**, dass die dritte Bitfolge aus einer jeweils aktuellen Datums- und/oder Zeitangabe generiert wird.

8. Verfahren nach Anspruch 6 oder 7, **dadurch gekennzeichnet**, dass die dritte Bitfolge zur Initialisierung einem dritten rückgekoppelten Schieberegister zugeführt wird.

9. Verfahren nach einem der Ansprüche 1 bis 8, **dadurch gekennzeichnet**, dass die Generierung des Schlüsselstroms beginnt, sobald wenigstens eines der rückgekoppelten Schieberegister mit dem ersten Bit aus der jeweiligen Bitfolge gefüllt wird.

10. Verfahren nach einem der Ansprüche 1 bis 9, **dadurch gekennzeichnet**, dass die rückgekoppelten Schieberegister gleichzeitig mit der jeweiligen Bitfolge gefüllt werden.

11. Verfahren nach einem der Ansprüche 1 bis 10, **dadurch gekennzeichnet**, dass zur Rückkoppelung des bzw. der Schieberegister wenigstens ein XOR-Gatter verwendet wird.

12. Verfahren nach einem der Ansprüche 1 bis 11, **dadurch gekennzeichnet**, 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.

13. Verfahren nach einem der Ansprüche 1 bis 12, **dadurch gekennzeichnet**, dass das wenigstens eine rückgekoppelte Schieberegister eine Mehrzahl von zu einer codeproduzierenden Reihe geschalteten Speicherelementen 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 derart erfolgt, dass der erster Eingang des XOR-Gatters mit dem Ausgang eines in der codeproduzierenden Reihe befindlichen Speicherelements, der zweite Eingang mit dem Ausgang eines weiteren in der codeproduzierenden Reihe befindlichen Speicherelements und der Ausgang mit dem Eingang des in der codeproduzierenden Reihe dem mit dem ersten Eingang des XOR-Gatters verbundenen Speicherelement nachfolgenden Speicherelements verbunden ist.

14. Verfahren nach Anspruch 13, **dadurch gekennzeichnet**, dass in die den zweiten Eingang des wenigstens einen XOR-Gatters und den Ausgang des weiteren in der codeproduzierenden Reihe befindlichen Speicherelements verbindende Leitung ein UND-Gatter derart geschalten ist, dass der Ausgang des UND-Gatters mit dem zweiten Eingang des XOR-Gatters, der erste Eingang des UND-Gatters mit dem Ausgang des weiteren in der codeproduzierenden Reihe befindlichen Speicherelements und der zweite Eingang des UND-Gatters mit dem Ausgang eines codeprogrammierenden Speicherelements verbunden ist, wobei als codeprogrammierendes Speicherelement ein Speicherelement eines weiteren rückgekoppelten Schieberegisters verwendet wird, und dass bevorzugt der Ausgang eines in der codeproduzierenden Reihe befindlichen Speicherelements mit dem Eingang eines Inverters und der Ausgang des Inverters mit dem Eingang eines anderen in der codeproduzierenden Reihe angeordneten Speicherelements verbunden ist.

15. Vorrichtung zum Verschlüsseln von Daten mit Hilfe einer symmetrischen Stromverschlüsselung unter Verwendung eines Schlüsselstroms (3), insbesondere zur Durchführung des Verfahrens nach einem der Ansprüche 1 und 3 bis 14, wobei zur Generierung des Schlüsselstroms (3) wenigstens ein erstes und ein zweites rückgekoppeltes Schieberegister (27; 30,31,32; 33,34; 35,36,37) vorgesehen sind, die zu ihrer Initialisierung jeweils mit einer definierten Bitfolge gefüllt werden, **dadurch gekennzeichnet**, dass die Daten in Datenpakete (1) aufgeteilt vorliegen, dass Mittel (9,10) zum Generieren und/oder Speichern wenigstens einer ersten Bitfolge (6) und einer zweiten Bitfolge (7) vorgesehen sind, die mit den Schieberegistern (27; 30,31,32; 33,34; 35,36,37) derart zusammenwirken, dass die erste Bitfolge (6) dem ersten rückgekoppelten Schieberegister (30;33;35) zu dessen Initialisierung zugeführt ist und die zweite Bitfolge (7) dem zweiten rückgekoppelten Schieberegister (31;34;36) zu dessen Initialisierung zugeführt ist, wobei die rückgekoppelten Schieberegister (27; 30,31,32; 33,34; 35,36,37) für die Verschlüsselung jedes Datenpakets (1) neu initialisiert werden, dass Datenpaketverarbeitungsmittel (15) vorgesehen sind, mit denen die Mittel (9,10) zum Generieren bzw. Speichern der ersten (6) und der zweiten (7) Bitfolge derart zusammenwirken, dass die erste Bitfolge (6) dem jeweils verschlüsselten Datenpaket im Klartext (17) oder in codierter Form hinzugefügt wird und die zweite Bitfolge (7) einen geheimen Schlüssel darstellt, die den verschlüsselten Datenpaketen nicht hinzugefügt wird, und dass Datenübertragungsmittel (19) zum paketvermittelten Versenden der verschlüsselten Datenpakete samt der jeweiligen hinzugefügten Bitfolge (17) und ggf. Kopfdaten (16) vorgesehen sind.

16. Vorrichtung zum Entschlüsseln von mittels einer symmetrischen Stromverschlüsselung verschlüsselten Daten unter Verwendung eines Schlüsselstroms (3), insbesondere zur Durchführung des Verfahrens nach einem der Ansprüche 2 bis 14, wobei zur Generierung des Schlüsselstroms (3) wenigstens ein erstes und ein zweites rückgekoppeltes Schieberegister (27; 30,31,32; 33,34; 35,36,37) vorgesehen sind, die zu ihrer Initialisierung jeweils mit einer definierten Bitfolge gefüllt werden, **dadurch gekennzeichnet**, dass die verschlüsselten Daten in Datenpakete (1) aufgeteilt vorliegen, dass Mittel (20) zum Auslesen einer ersten Bitfolge (6) im Klartext oder in codierter Form aus den Datenpaketen und Mittel (24) zum Generieren und/oder Speichern wenigstens einer zweiten Bitfolge (7) vorgesehen sind, die mit den Schieberegistern (27; 30,31,32; 33,34; 35,36,37) derart zusammenwirken, dass die erste Bitfolge (6) dem ersten rückgekoppelten Schieberegister (30;33;35) zu dessen Initialisierung zugeführt ist und die zweite Bitfolge (7) dem zweiten rückgekoppelten Schieberegister (31;34;36) zu dessen Initialisierung zugeführt ist, wobei die rückgekoppelten Schieberegister (27; 30,31,32; 33,34; 35,36,37) für die Entschlüsselung jedes Datenpakets neu initialisiert werden, wobei die zweite Bitfolge (7) einen geheimen Schlüssel darstellt, die aus den verschlüsselten Datenpaketen nicht ausgelesen werden kann.

17. Vorrichtung nach Anspruch 15 oder 16, **dadurch gekennzeichnet**, dass die erste Bitfolge (6) eine für das zu verschlüsselnde Datenpaket (1) eindeutige Bitfolge ist, die dem jeweils verschlüsselten Datenpaket als Paketkennung (17) im Klartext oder in codierter Form hinzugefügt ist.

18. Vorrichtung nach einem der Ansprüche 15 bis 17, **dadurch gekennzeichnet**, dass Mittel (13;25) zum Generieren der zweiten Bitfolge (7) aus einer eindeutigen Kennung (11) des Senders und einer eindeutigen Kennung (12) des Empfängers vorgesehen sind.

19. Vorrichtung nach Anspruch 18, **dadurch gekennzeichnet**, dass die Mittel (13;25) zum Generieren der zweiten Bitfolge (7) ein XOR-Gatter umfasst, dessen einem Eingang die eindeutige Kennung (11) des Senders und dessen anderem Eingang die eindeutige Kennung (12) des Empfängers zugeführt ist.

20. Vorrichtung nach einem der Ansprüche 15 bis 19, **dadurch gekennzeichnet**, dass Mittel (14;26) zum Generieren und/oder Speichern wenigstens einer dritten Bitfolge (8) vorgesehen sind, die mit dem bzw. den Schieberegister(n) (27;32;37) derart zusammenwirken, dass auch die dritte Bitfolge (8) zur Initialisierung des bzw. der rückgekoppelten Schieberegister (27;32;37) verwendet wird.

21. Vorrichtung nach Anspruch 20, **dadurch gekennzeichnet**, dass die dritte Bitfolge (8) aus einer jeweils aktuellen Datums- und/oder Zeitangabe generiert wird.

22. Vorrichtung nach Anspruch 20 oder 21, **dadurch gekennzeichnet**, dass die dritte Bitfolge (8) zur Initialisierung einem dritten rückgekoppelten Schieberegister (32;37) zugeführt ist.

23. Vorrichtung nach einem der Ansprüche 15 bis 22, **dadurch gekennzeichnet**, dass die Generierung des Schlüsselstroms (3) beginnt, sobald wenigstens eines der rückgekoppelten Schieberegister (27; 30,31,32; 33,34; 35,36,37) mit dem ersten Bit aus der jeweiligen Bitfolge gefüllt wird.

24. Vorrichtung nach einem der Ansprüche 15 bis 23, **dadurch gekennzeichnet**, dass die rückgekoppelten Schieberegister (30,31,32; 33,34; 35,36,37) gleichzeitig mit der jeweiligen Bitfolge gefüllt werden.

25. Vorrichtung nach einem der Ansprüche 15 bis 24, **dadurch gekennzeichnet**, dass zur Rückkoppelung des bzw. der Schieberegister (27; 30,31,32; 33,34; 35,36,37) wenigstens ein XOR-Gatter (XORp1, XORp2, XORp3, XORp4, XORpp1, XORppp1) eingesetzt ist.

26. Vorrichtung nach einem der Ansprüche 15 bis 25, **dadurch gekennzeichnet**, dass die rückgekoppelten Schieberegister (30,31,32; 33,34; 35,36,37) 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.

27. Vorrichtung nach einem der Ansprüche 15 bis 26, **dadurch gekennzeichnet**, dass das wenigstens eine rückgekoppelte Schieberegister (30,31,32; 33,34; 35,36,37) eine Mehrzahl von zu einer codeproduzierenden Reihe geschalteten Speicherelementen (FF1, FF 2, ...; 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.

28. Vorrichtung nach Anspruch 27, **dadurch gekennzeichnet**, dass in die den zweiten Eingang des wenigstens einen XOR-Gatters (XORp1) und den Ausgang des weiteren in der codeproduzierenden Reihe (30;33;35) 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 (30;33;35) 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 (30;33;35) befindlichen Speicherelements (FF9) mit dem Eingang eines Inverters (INV) und der Ausgang des Inverters (INV) mit dem Eingang eines anderen in der codeproduzierenden Reihe (30;33;35) angeordneten Speicherelements (FF1) verbunden ist, wobei als codeprogrammierendes Speicherelement ein Speicherelement eines weiteren rückgekoppelten Schieberegisters (31;34;36) verwendet wird.

29. Vorrichtung nach Anspruch 27 oder 28, **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 (30;33;35) befindlichen Speicherelements (FF1,2,3,4) gespeist wird und deren zweiter Eingang jeweils vom Ausgang eines weiteren in der codeproduzierenden Reihe (30;33;35) befindlichen Speicherelements (FF8,15,20,23) gespeist wird, welches eine Anzahl von Speicherelementen in Flussrichtung der Reihe (30;33;35) 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 (30;33;35) geschalteten Speicherelemente (FF1,2,...n) ist.

30. Vorrichtung nach einem der Ansprüche 27 bis 29, **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 (31;34;36) geschalten ist und wenigstens ein XOR-Gatter (XORpp1) angeordnet ist, dessen erster Eingang mit dem Ausgang eines in der codeprogrammierenden Reihe (31;34;36) befindlichen Speicherelements (FFp6), dessen zweiter Eingang mit dem Ausgang eines weiteren in der codeprogrammierenden Reihe (31;34;36) befindlichen Speicherelements (FFp5) und dessen Ausgang mit dem Eingang des in der codeprogrammierenden Reihe (31;34;36) dem mit dem ersten Eingang des XOR-Gatters (XORpp1) verbundenen Speicherelement (FFp6) nachfolgenden Speicherelements (FFp1) verbunden ist.

31. Vorrichtung nach einem der Ansprüche 27 bis 30, **dadurch gekennzeichnet**, dass in die den zweiten Eingang des wenigstens einen XOR-Gatters (XORpp1) und den Ausgang des weiteren in der codeprogrammierenden Reihe (31;34;36) 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 (31;34;36) befindlichen Speicherelements (FFp3) und der zweite Eingang des UND-Gatters (UNDpp1) mit dem Ausgang eines der Programmierung der codeprogrammierenden Reihe (31;34;36) dienenden Speicherelements (FFpp5) verbunden ist.

32. Vorrichtung nach einem der Ansprüche 27 bis 31, **dadurch gekennzeichnet**, dass eine Mehrzahl von der Programmierung der codeprogrammierenden Reihe (31;34;36) 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 (32;37) geschalten ist und wenigstens ein XOR-Gatter (XORppp1) angeordnet ist, dessen erster Eingang mit dem Ausgang eines in der Reihe (32;37) befindlichen Speicherelements (FFpp1), dessen zweiter Eingang mit dem Ausgang eines weiteren in der Reihe (32;37) befindlichen Speicherelements (FFpp3) und dessen Ausgang mit dem Eingang des in der Reihe (32;37) dem mit dem ersten Eingang des XOR-Gatters (XORppp1) verbundenen Speicherelement (FFpp1) nachfolgenden Speicherelements (FFpp2) verbunden ist.

Hierzu 8 Blatt Zeichnungen

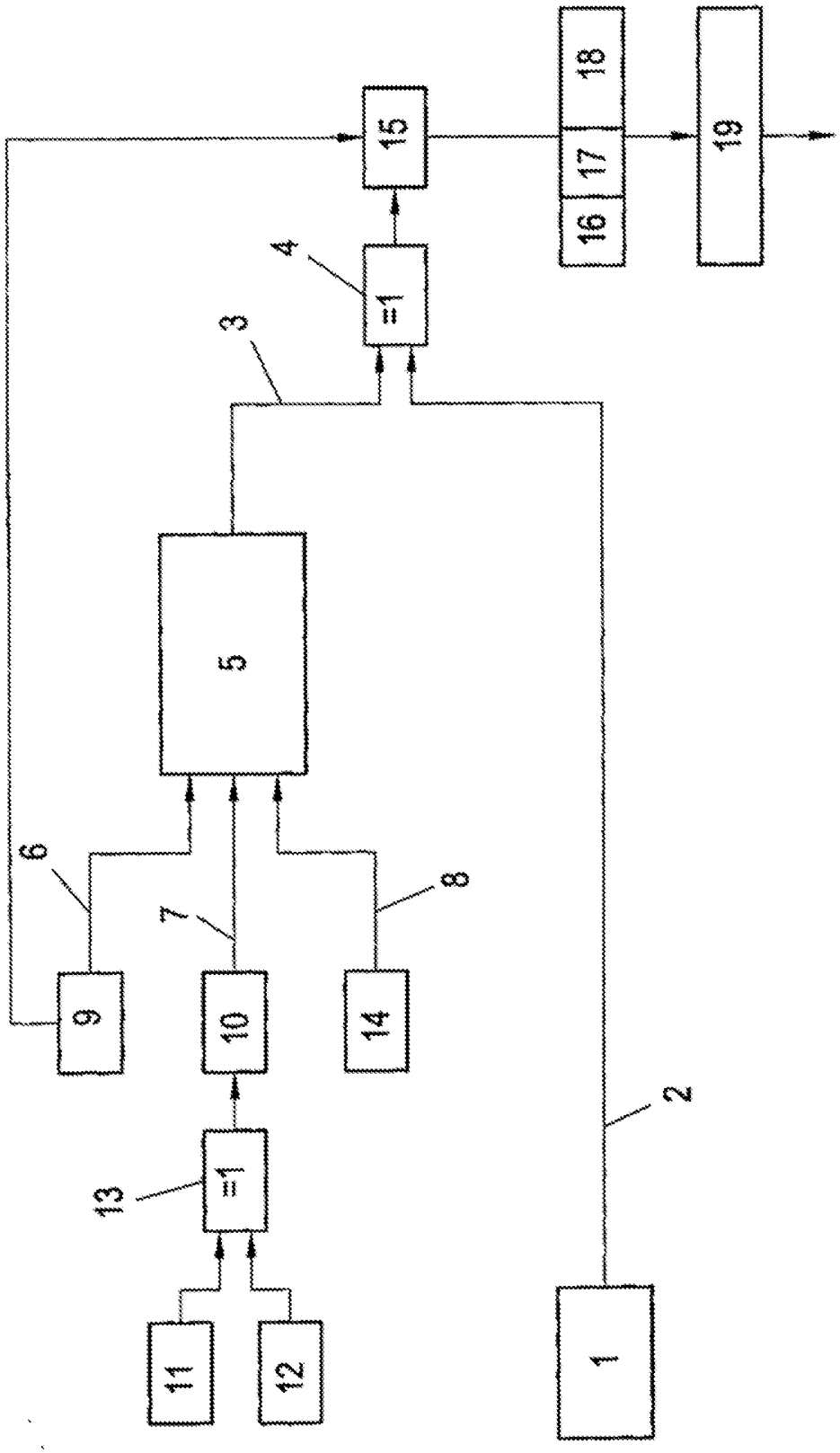


Fig. 1

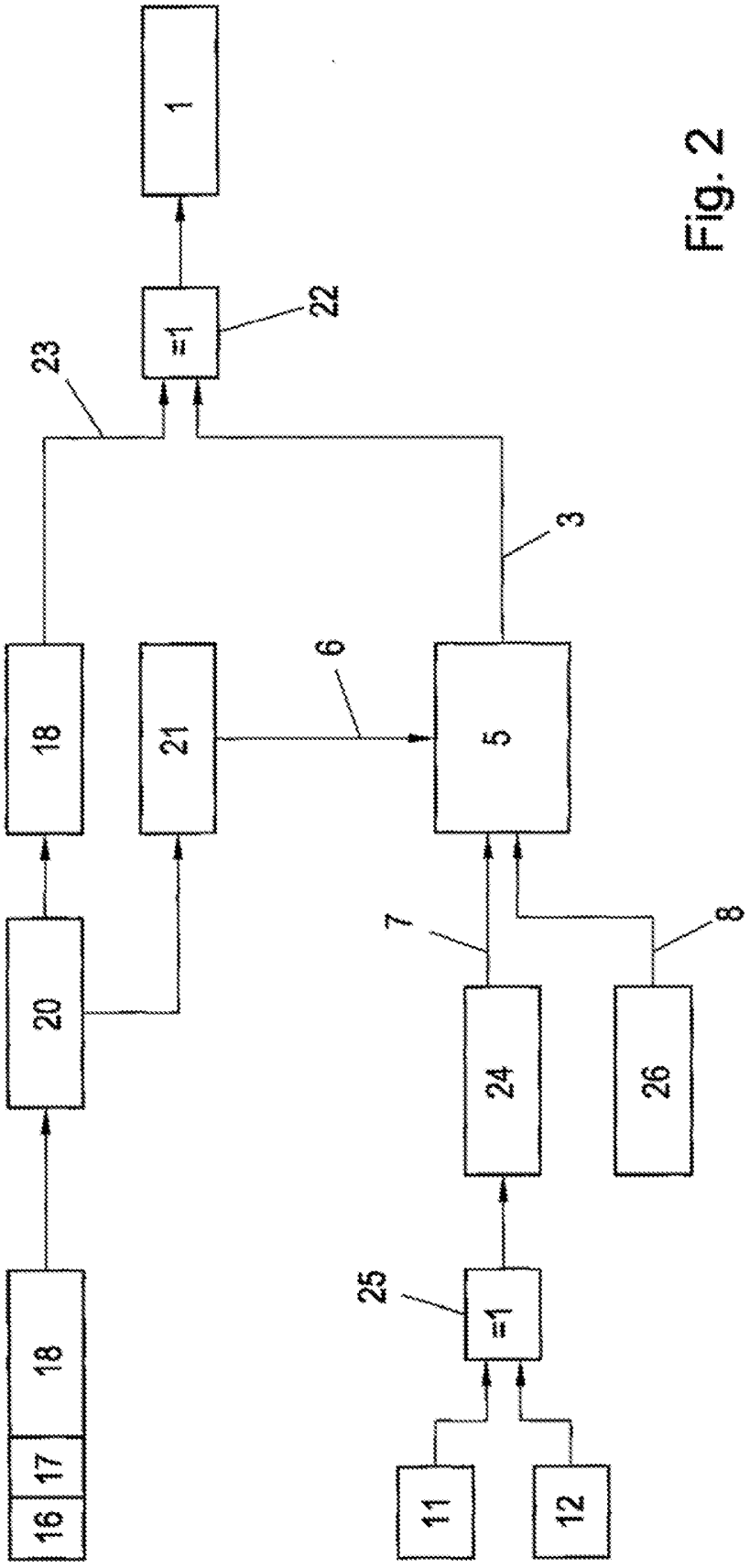


Fig. 2

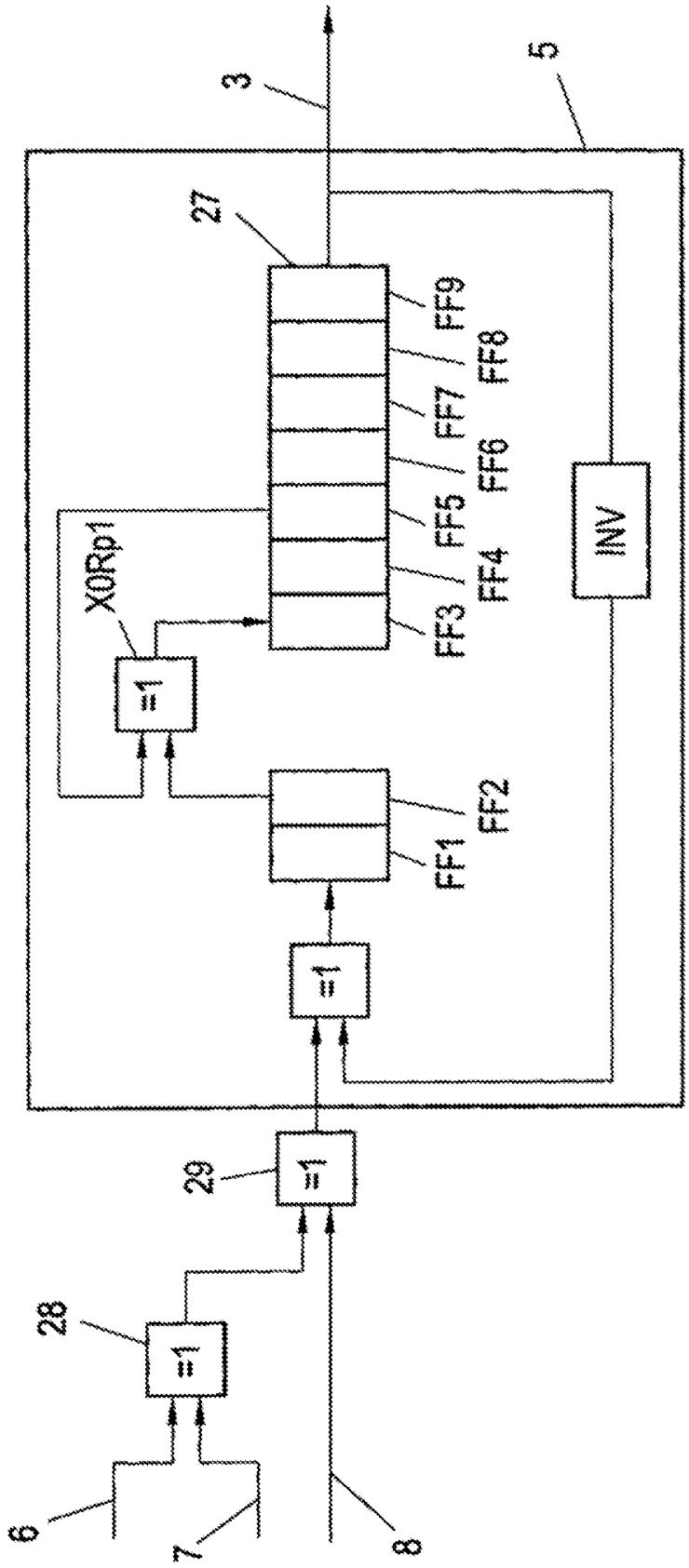

3
5
27
FF9
FF8
FF7
FF6
FF5
FF4
FF3
FF2
FF1
INV
X0Rp1
=1
=1
=1
=1
28
29
6
7
8

Fig. 3

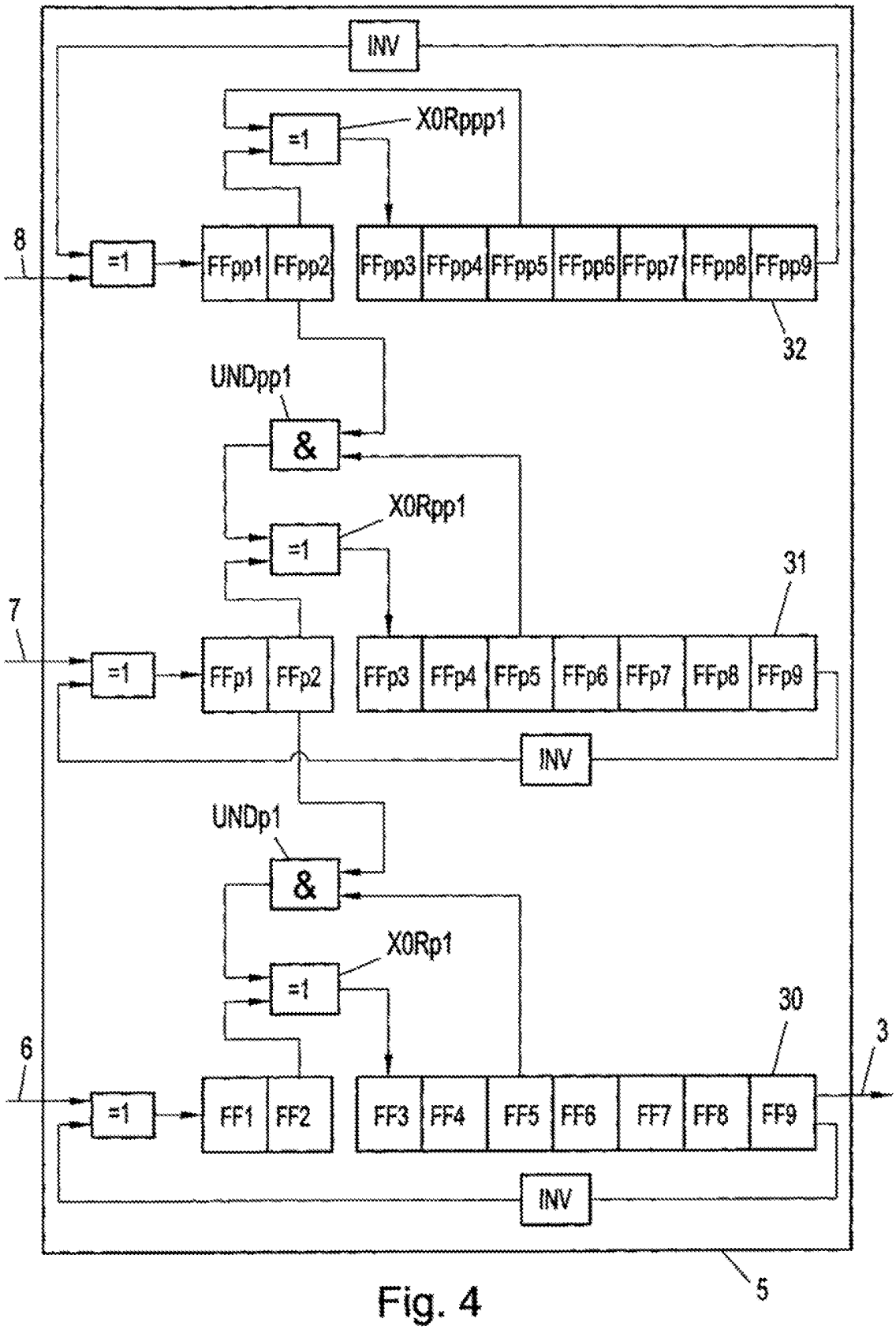

INV
=1
X0Rppp1
8
=1
FFpp1
FFpp2
FFpp3
FFpp4
FFpp5
FFpp6
FFpp7
FFpp8
FFpp9
32
UNDpp1
&
X0Rpp1
=1
31
7
=1
FFp1
FFp2
FFp3
FFp4
FFp5
FFp6
FFp7
FFp8
FFp9
INV
UNDp1
&
X0Rp1
=1
30
3
6
=1
FF1
FF2
FF3
FF4
FF5
FF6
FF7
FF8
FF9
INV
5

Fig. 4

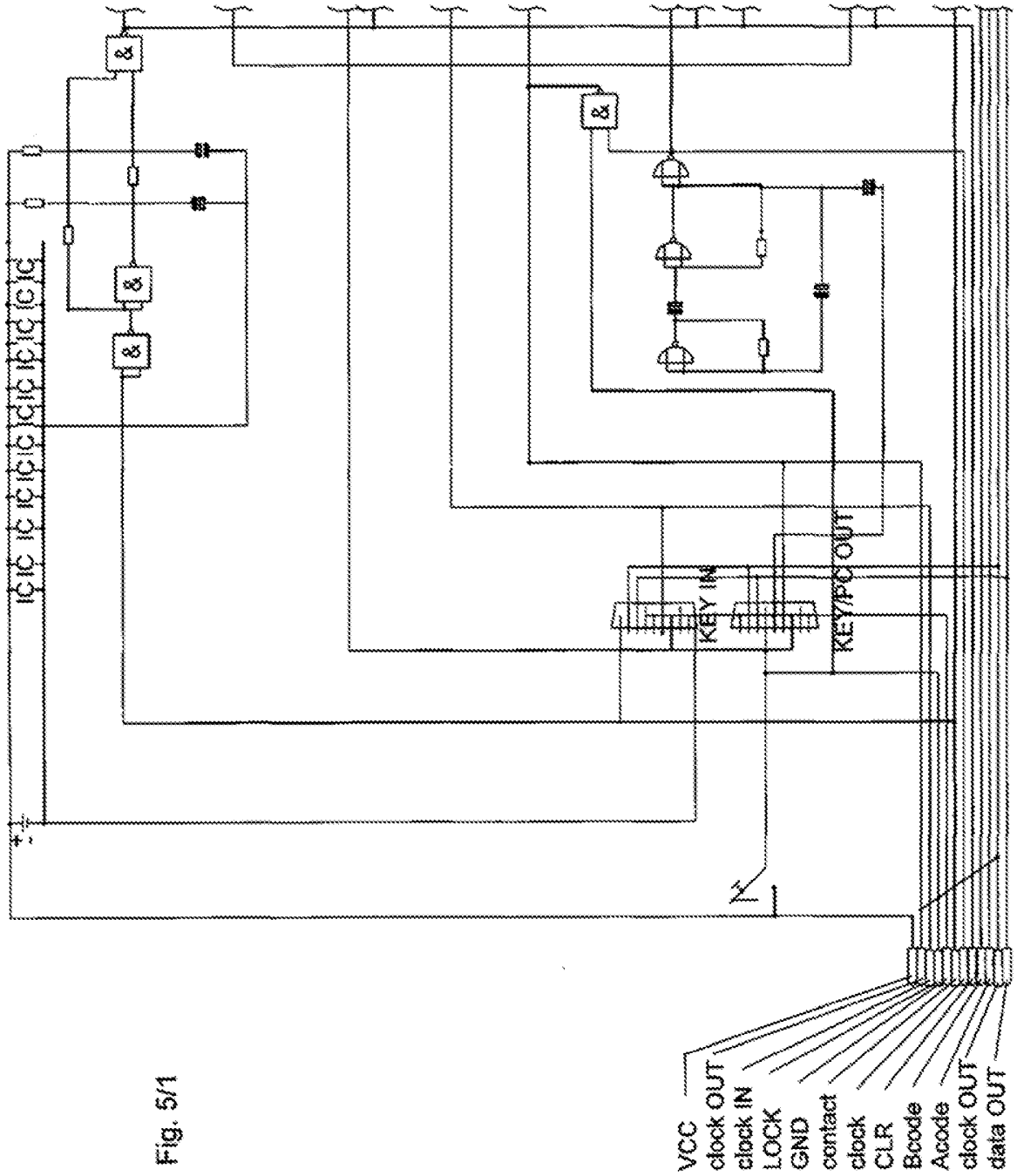
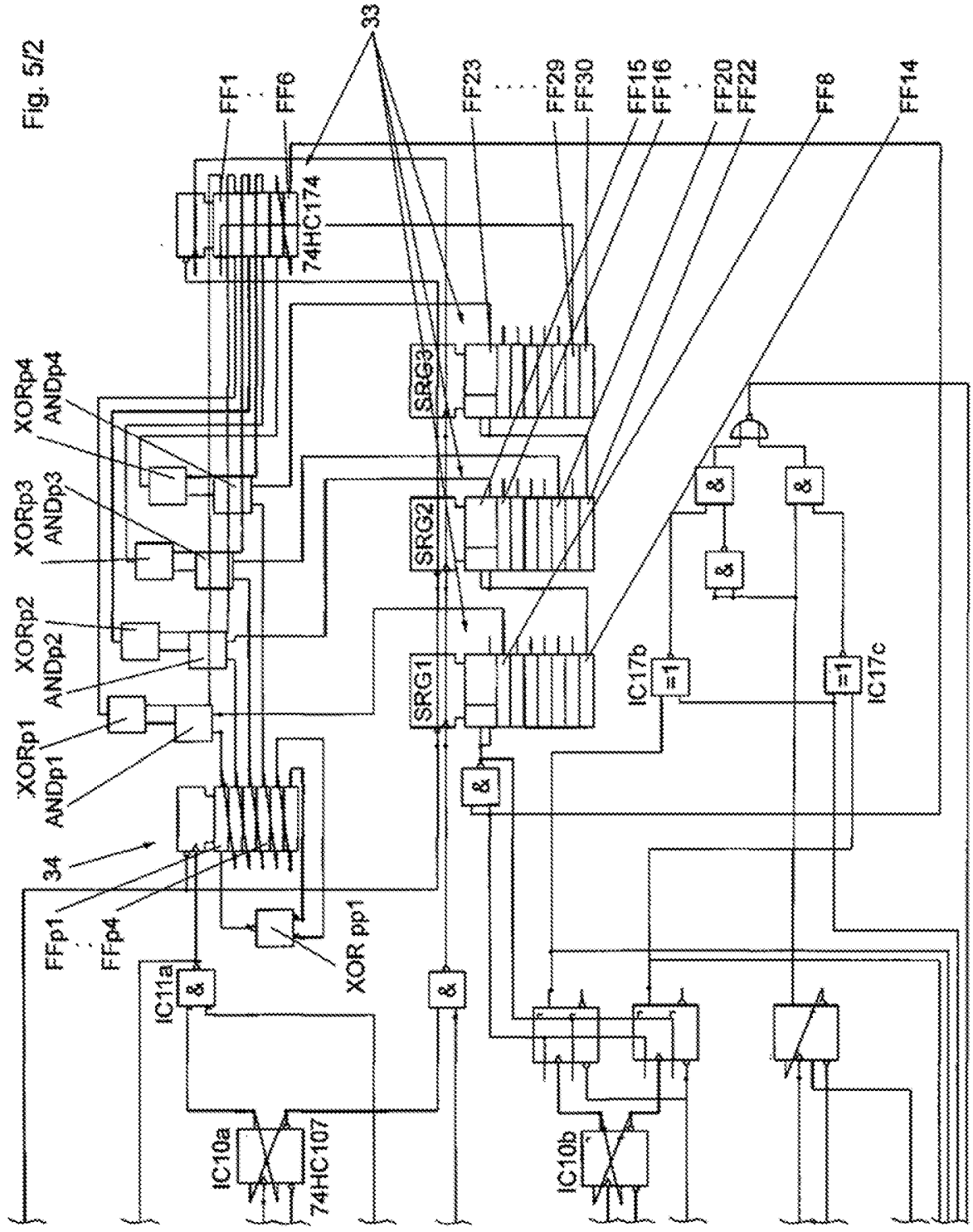

&
&
&
&
IC IC IC IC IC IC IC IC IC IC IC IC
KEY IN
KEY/PC OUT
VCC
clock OUT
clock IN
LOCK
GND
contact
clock
CLR
Bcode
Acode
clock OUT
data OUT

Fig. 5/1

Fig. 5/2


33
FF1
FF6
FF23
FF29
FF30
FF15
FF16
FF20
FF22
FF8
FF14
74HC174
SRG3
SRG2
SRG1
XORp4
ANDp4
XORp3
ANDp3
XORp2
ANDp2
XORp1
ANDp1
34
FFp1
FFp4
XOR pp1
IC11a
IC10a
74HC107
IC10b
IC17b
IC17c
=1
&

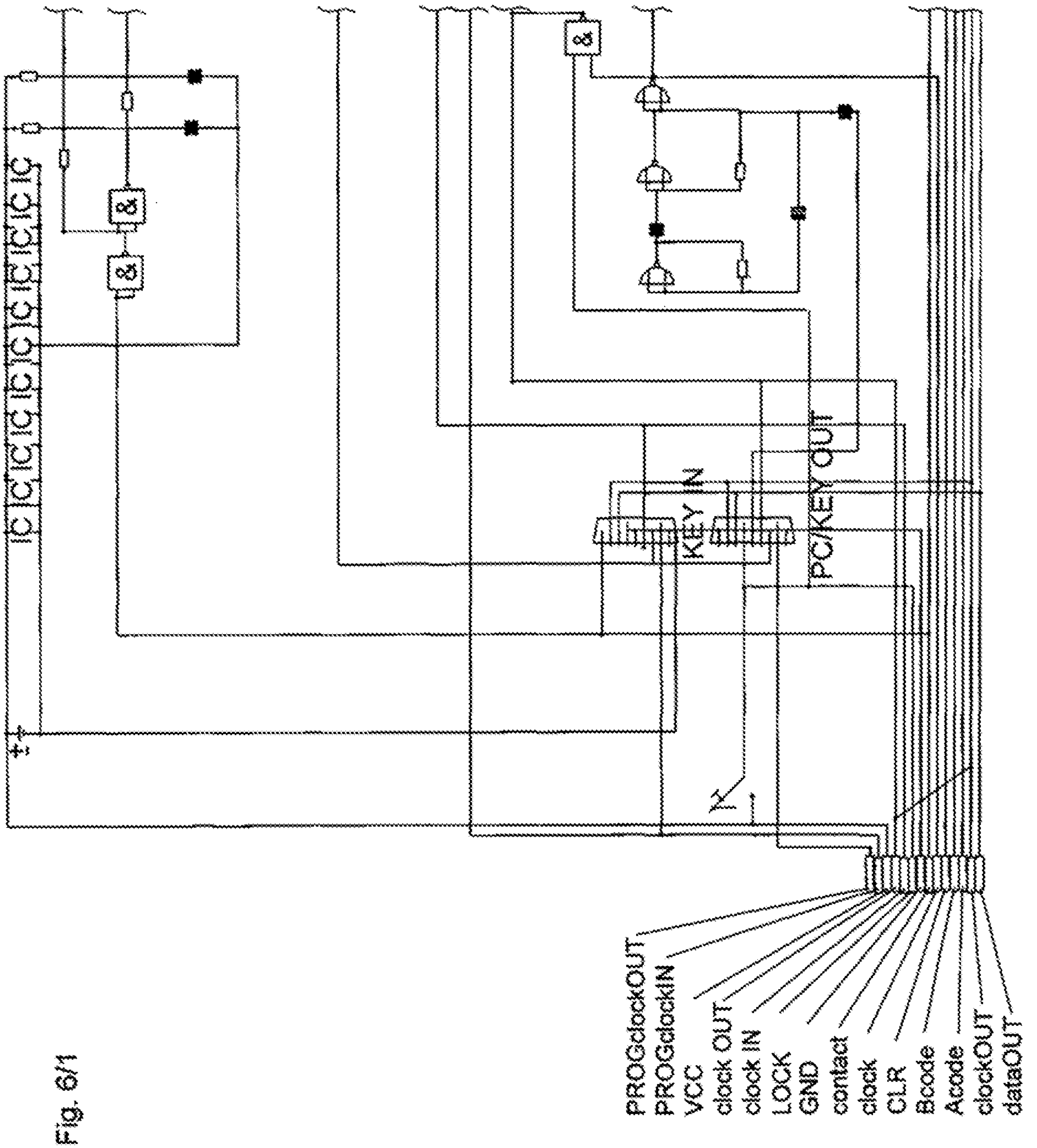
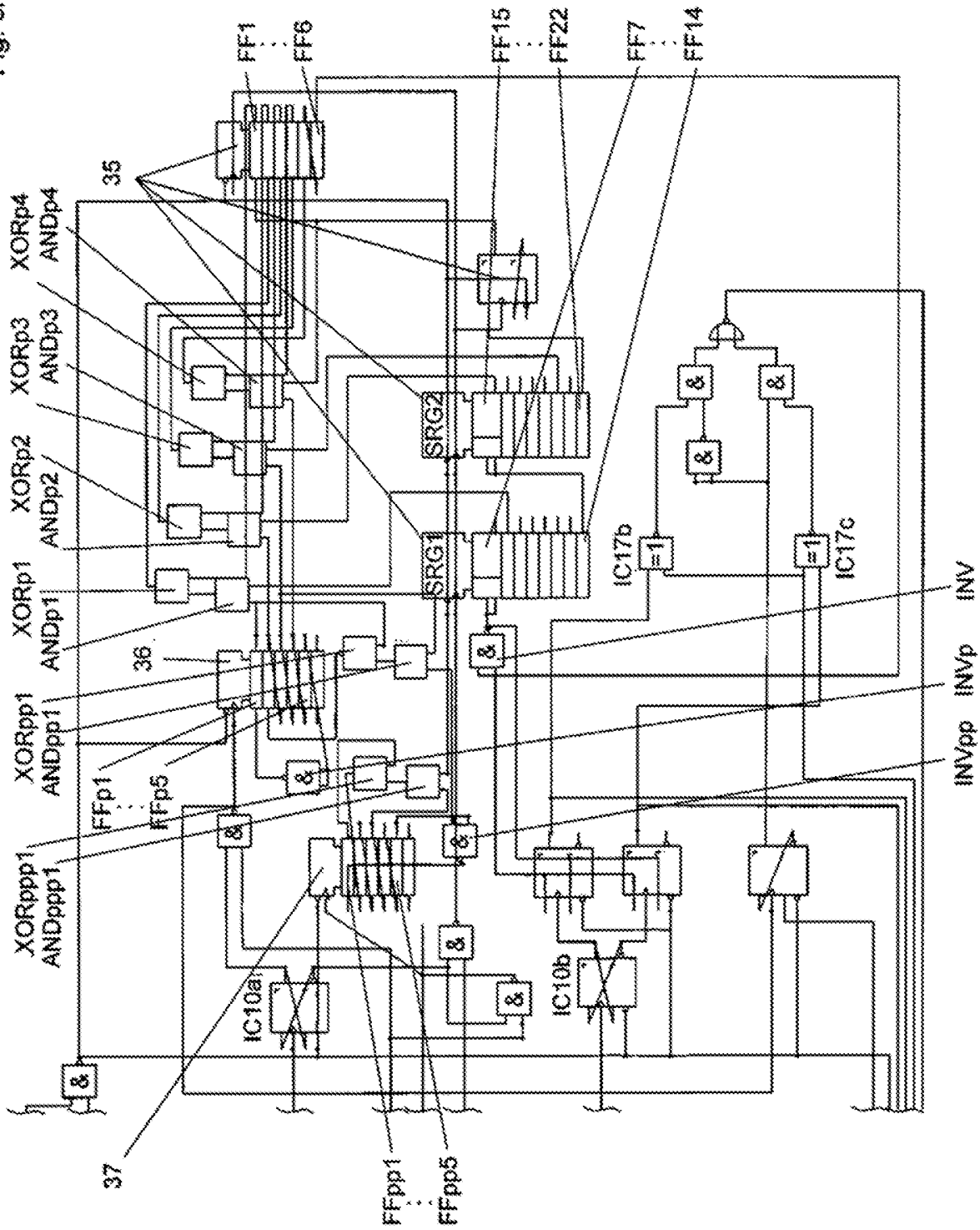

IC IC IC IC IC IC IC IC IC IC IC IC
&
&
&
KEY IN
PC/KEY OUT
PROGclockOUT
PROGclockIN
VCC
clock OUT
clock IN
LOCK
GND
contact
clock
CLR
Bcode
Acode
clockOUT
dataOUT

Fig. 6/1

Fig. 6/2


FF1
FF6
FF15
FF22
FF7
FF14
35
XORp4
ANDp4
XORp3
ANDp3
XORp2
ANDp2
XORp1
ANDp1
XORpp1
ANDpp1
XORppp1
ANDppp1
SRG2
SRG1
IC17b
IC17c
=1
INV
INVp
INVpp
36
FFp1
FFp5
IC10a
IC10b
37
FFpp1
FFpp5