



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 699 06 022 T2** 2004.01.08

(12)

Übersetzung der europäischen Patentschrift

(97) **EP 1 011 081 B1**

(21) Deutsches Aktenzeichen: **699 06 022.2**

(96) Europäisches Aktenzeichen: **99 124 934.3**

(96) Europäischer Anmeldetag: **14.12.1999**

(97) Erstveröffentlichung durch das EPA: **21.06.2000**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **19.03.2003**

(47) Veröffentlichungstag im Patentblatt: **08.01.2004**

(51) Int Cl.⁷: **G07F 7/10**
G06K 19/073

(30) Unionspriorität:

35415698 14.12.1998 JP

(73) Patentinhaber:

Hitachi, Ltd., Tokio/Tokyo, JP

(74) Vertreter:

Strehl, Schübel-Hopf & Partner, 80538 München

(84) Benannte Vertragsstaaten:

DE, FR, GB

(72) Erfinder:

Oki, Masaru, Tokyo, JP; Fukuzawa, Yasuko, Tokyo, JP; Okuhara, Susumu, Tokyo, JP; Kaminaga, Masahiro, Tokyo, JP

(54) Bezeichnung: **Datenverarbeitungsanlage**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Die vorliegende Erfindung betrifft eine Informationsverarbeitungseinrichtung, und insbesondere ein Informationsverarbeitungsgerät, das für eine fälschungssichere Vorrichtung geeignet ist, wie zum Beispiel eine IC-Karte, die hohe Sicherheit bietet.

[0002] Eine IC-Karte wird hauptsächlich dazu verwendet, Informationen auf solche Weise zu speichern, dass sie nicht durch Dritte verändert werden können, oder zum Verschlüsseln von Daten oder Entschlüsseln eines verschlüsselten Textes, unter Verwendung eines Verschlüsselungscodes, welcher geheim gehalten wird. Da die IC-Karte nicht über eine Stromquelle verfügt, kann sie nur betrieben werden, wenn sie in ein Lese-/Schreibgerät eingeführt ist. Die IC-Karte empfängt von dem Lese-/Schreibgerät einen Befehl, um Datenübertragung auszuführen.

[0003] Wie in **Fig. 1** dargestellt, weist eine IC-Karte den Aufbau auf, dass ein IC-Kartenchip **102** auf einer Karte **101** hergestellt ist. Eine gebräuchliche IC-Karte weist Kontakte auf, über welche Strom von dem Lese-/Schreibgerät geliefert wird und Daten übertragen werden.

[0004] Der Aufbau eines IC-Kartenchips ist im Wesentlichen derselbe wie derjenige eines Mikrocomputers. Wie in **Fig. 2** dargestellt, umfasst der IC-Kartenchip einen zentralen Prozessor **201**, einen Speicher **204**, einen Eingabe-/Ausgabeanschluss **207**, und einen Co-Prozessor **202**. Der zentrale Prozessor **201** führt logische und arithmetische Berechnungen aus, und der Speicher **204** speichert Programme und Daten. Der Eingabe-/Ausgabeanschluss **207** kommuniziert mit einem Lese-/Schreibgerät. Der Co-Prozessor **202** ist eine spezielle Rechenvorrichtung zum Ausführen von Modularrechnungen und wird für die Berechnungen von antisymmetrischen RSA oder dergleichen verwendet. Viele IC-Kartenprozessoren weisen keinen Co-Prozessor auf. Ein Datenbus **203** verbindet die Komponenten der IC-Karte untereinander.

[0005] Der Speicher **204** umfasst ein ROM (Festwertspeicher), ein RAM (Direktzugriffsspeicher), ein EEPROM (elektrisch löschbarer programmierbarer Festwertspeicher) und dergleichen. ROM ist eine Speichervorrichtung, deren Inhalte nicht frei überschreibbar sind, und wird hauptsächlich für das Speichern von Programmen verwendet. RAM ist ein Speicher, dessen Inhalte frei überschrieben und gelöscht werden können, wenn eine Stromversorgung unterbrochen wird. Wenn die IC-Karte von dem Lese-/Schreibgerät getrennt wird, wird eine Versorgung mit Strom unterbrochen, so dass die Inhalte von RAM nicht erhalten werden können. EEPROM ist ein Speicher, dessen Inhalte erhalten werden können, selbst wenn eine Stromversorgung unterbrochen wird. Deswegen wird EEPROM zum Speichern von Daten verwendet, welche überschrieben werden können und erhalten werden können, selbst wenn die IC-Karte von dem Lese-/Schreibgerät getrennt wird. Zum Bei-

spiel ist die Anzahl der im Voraus bezahlten Zeit einer vorausbezahlten Karte in EEPROM gespeichert, weil sie jedes Mal aktualisiert wird, wenn die Karte benutzt wird, und es erforderlich ist, dass ihre Inhalte erhalten werden, selbst wenn die Karte von dem Lese-/Schreibgerät getrennt ist.

[0006] Eine IC-Karte wird zum Speichern von Programmen und wichtigen Informationen in dem IC-Kartenchip verwendet, um einen Verschlüsselungsprozess auszuführen. Man ist lange der Meinung gewesen, dass die Schwierigkeit beim Entschlüsseln eines Verschlüsselungsprozesses, der in der IC-Karte ausgeführt wird, dieselbe ist wie diejenige beim Entschlüsseln eines Verschlüsselungsalgorithmus. Kürzlich ist jedoch darauf hingewiesen worden, dass es eine Möglichkeit gibt, die Inhalte eines Verschlüsselungsprozesses und eines Verschlüsselungscodes abzuleiten, indem, während der Ausführung des Verschlüsselungsprozesses, ein Verbrauchsstrom gemessen und analysiert wird, was einfacher ist als die Entschlüsselung eines Verschlüsselungsalgorithmus. Der Verbrauchsstrom kann durch Messen des Stroms, der von dem Lese-/Schreibgerät geliefert wird, überwacht werden. Diese mögliche Gefahr ist in „Smart Card Handbook“ (Handbuch der intelligenten Karte), von W. Rankl & W. Effing, John Wiley & Sons, Paragraph 8.5.1.1 „Passive protective mechanism“ (Passiver Schutzmechanismus), auf S. 263 beschrieben.

ZUSAMMENFASSUNG DER ERFINDUNG

[0007] CMOS, die einen IC-Kartenchip bilden, verbrauchen Strom, wenn ein Ausgabezustand von „1“ auf „0“ oder umgekehrt wechselt. Insbesondere fließt in dem Datenbus **203** eine große Menge Strom, wenn sein Zustand von „1“ auf „0“ oder umgekehrt wechselt, weil der Datenbus eine große elektrische Kapazität aufweist. Daher besteht eine Möglichkeit, den Operationszustand in dem IC-Kartenchip durch Überwachen des Verbrauchsstroms abzuleiten.

[0008] **Fig. 3** stellt Schwingungsformen von Verbrauchsstrom während eines Operationszyklus eines IC-Kartenchips dar. Je nachdem welche Daten verarbeitet werden, verändert sich die Stromschwingungsform, wie bei **301** und **302** angezeigt. Dieser Unterschied wird in Abhängigkeit von Daten auf dem Bus **203** und von Daten, die durch den zentralen Prozessor **201** verarbeitet werden, erzeugt.

[0009] Betrachten wir nun die Datenübertragung auf einem Vor-Ladungs-Bus von 16 Bits. Der Vor-Ladungs-Bus wird vor der Datenübertragung zurückgesetzt, so dass alle Bits auf dem Bus einen Wert „0“ haben. Wenn die Daten, die dieselbe Anzahl von „1“ Bits und verschiedene Werte aufweisen, z. B. Hexadezimaldaten mit „88“ und „11“, die beide zwei „1“ Bits aufweisen, zu diesem Bus übertragen werden, sind die Stromschwingungsformen im Allgemeinen dieselben. Dieses ist so, weil die Anzahl an Bits, die von „0“ auf „1“ wechseln, dieselbe ist und derselbe

Strom verbraucht wird, um ähnliche Stromschwingungsformen aufzuweisen. Wenn die Daten, die einen Unterschied von einem „1“ Bit aufweisen, z. B. Hexadezimaldaten mit „89“ und „19“, die beide drei „1“ Bits aufweisen, zu diesem Bus übertragen werden, wird die Stromschwingungsform anders als diejenige der Daten, die zwei „1“ Bits aufweisen. Dieses ist so, weil die Anzahl der Bits, die von „0“ auf „1“ wechseln, auf drei Bits wechselt und Strom entsprechend ansteigend verbraucht wird. Deswegen, wie verglichen mit den Daten, die zwei „1“ Bits aufweisen, steigt der Stromverbrauch um eine Menge an, die einem Bit entspricht. Es besteht die Regelmäßigkeit, dass je größer die Anzahl von „1“ Bits, desto größer die Amplitude der Stromschwingungsform. Die übertragenen Daten können durch diese Regelmäßigkeit abgeleitet werden.

[0010] Die in **Fig. 3** dargestellten Stromschwingungsformen stellen die gesamte Summe an Strom dar, der nicht nur durch den Bus fließt, sondern auch durch andere Komponenten, die den IC-Kartenchip bilden. Ein Mikrocomputer, wie ein IC-Kartenchip, umfasst eine Phase, in welcher Daten hauptsächlich zu dem Bus übertragen werden, eine Phase, in welcher eine CPU hauptsächlich in Betrieb ist, eine Phase, in welcher Daten in ein Register geschrieben werden, und andere Phasen. Wenn diese Phasen berücksichtigt werden, ist es möglich herauszufinden, durch welche Komponente ein Unterschied in den Verbrauchsströmen hauptsächlich produziert wurde, und der Datenprozess an der Komponente kann abgeleitet werden.

[0011] Ein Unterschied in den Verbrauchsströmen wird beschrieben, indem der folgende Befehl Links-Verschieben als ein Beispiel verwendet wird.
Shift1 R1

(1)

[0012] Dieser Befehl verschiebt die Inhalte des Registers R1 nach links, d. h. verschiebt die Bitfolge in dem Register nach links, und der Wert des höchstwertigen Bits wird in ein Bedingungsanzeigeregister als ein Übertragbit eingegeben. Weil das höchstwertige Bit in dem Register R1 über den Datenbus zu dem Bedingungsanzeigeregister übertragen wurde, kann möglicherweise durch Vergleichen der Amplituden der Stromschwingungsformen unterschieden werden, ob das höchstwertige Bit „0“ oder „1“ ist. Wenn wichtige Daten in dem Register R1 gespeichert sind, gibt es eine Möglichkeit zu unterscheiden, ob diese Daten „0“ oder „1“ sind, obwohl die Daten nur aus einem Bit bestehen. In dem Verschlüsselungsprozess, insbesondere DES, wird häufig eine Operation des Verschiebens eines Verschlüsselungscodes verwendet. Während dieser Verschiebeoperation wird die Stromschwingungsform, die ermöglicht die Daten des Verschlüsselungscodes abzuleiten, so erzeugt, dass dabei eine Möglichkeit besteht, den Verschlüsselungscode abzuleiten.

[0013] Der oben beschriebene Fall wird ebenfalls

auf die Operation des Co-Prozessors **202** angewendet. Wenn die Operationsinhalte irgendeine Unsymmetrie enthalten, bedingt durch einen Verschlüsselungscode, kann diese Verschiebung von dem Verbrauchsstrom abgeleitet werden, und es besteht eine Möglichkeit, dass der Verschlüsselungscode abgeleitet wird.

[0014] Ein Problem, das mit der vorliegenden Erfindung zusammenhängt, ist die Verbindung zwischen dem Datenprozess in einem IC-Kartenchip und seinem Verbrauchsstrom zu reduzieren. Wenn die Verbindung zwischen dem Datenprozess in einem IC-Kartenchip und seinem Verbrauchsstrom reduziert werden kann, wird es schwierig, den Datenprozess in dem IC-Kartenchip und den Verschlüsselungscode von den beobachteten Verbrauchsstromformen her abzuleiten. Das Merkmal dieser Erfindung ist, Ableitung des Datenprozesses und des Verschlüsselungscodes von der Schwingungsform des Verbrauchsstroms schwierig zu gestalten, indem die Daten in dem IC-Kartenchip verarbeitet werden, nachdem sie umgewandelt wurden.

[0015] Diese Aufgaben werden nach der Erfindung durch die Merkmale der Ansprüche 1, 2, 3, 4 und 13 erfüllt.

[0016] Die fälschungssichere Vorrichtung, typischerweise ein IC-Kartenchip, wird als eine Informationsverarbeitungseinrichtung angesehen, welche umfasst: einen Speicher mit einer Programmspeichereinheit zum Speichern eines Programms und einer Datenspeichereinheit zum Speichern von Daten; und eine Zentralverarbeitungseinheit zum Ausführen eines Datenprozesses in Übereinstimmung mit dem Programm, wobei das Programm eine oder mehrere Datenverarbeitungseinrichtungen umfasst, die jeweils ein Prozessbefehl zum Erteilen eines Ausführungsbefehls an die Zentralverarbeitungseinheit sind. Nach der Erfindung, als das Verfahren zur Reduzierung der Verbindung zwischen dem Datenprozess in einem IC-Kartenchip und seinem Verbrauchsstrom, werden die Daten zuerst unter Verwendung von Stördaten umgewandelt und dann verarbeitet. Nach diesem Prozess werden die Daten unter Verwendung der Stördaten zurückgewandelt, um ein korrektes Verarbeitungsergebnis zu erhalten. Die Stördaten, die nach dem Datenprozess verwendet werden sollen, können, wenn erforderlich, dieselben Stördaten sein, die für den Datenprozess verwendet wurden. Die Stördaten werden bei jedem Datenprozess zufällig verändert. Durch diese Prozesse können, jeweils während der Datenprozesse, umgewandelte Daten verwendet werden, ohne die ursprünglichen Daten zu verwenden. Es wird deswegen schwierig, die Daten von den Stromschwingungsformen abzuleiten.

[0017] Insbesondere werden Stördaten Xi erst erzeugt und die Daten D1 werden unter Verwendung der Stördaten Xi umgewandelt, um umgewandelte Daten H1 zu erzeugen. Das Umwandlungsverfahren kann ein logisches Exklusiv-ODER, Addition, Multiplikation, oder dergleichen sein. Während des Daten-

prozesses werden die umgewandelten Daten H1 verarbeitet, um verarbeitete und umgewandelte Daten H2 zu erzeugen. Weil die umgewandelten Daten H1 anstelle von ursprünglichen Daten D1 verwendet werden, ist es schwierig, die Daten D1 von den Stromschwingungsformen während des Verarbeitens der umgewandelten Daten H1 abzuleiten. Weil die umgewandelten Daten unter Verwendung von unterschiedlichen Stördaten Xi bei jedem Prozess erzeugt werden, sind die umgewandelten Daten, die bei jedem Prozess erzeugt werden, unterschiedlich. Demzufolge verändert sich die Stromschwingungsform während des Verarbeitens der umgewandelten Daten H1 bei jedem Prozess. Das Ableiten der umgewandelten Daten H1 von den Stromschwingungsformen wird demzufolge bedeutungslos.

[0018] Wenn es notwendig ist, die Stördaten Xi in einer ähnlichen Weise wie die Daten D1 verarbeiten zu können, werden die Stördaten Xi zur Erzeugung von verarbeiteten Stördaten verarbeitet. Die verarbeiteten und umgewandelten Daten H1 werden unter Verwendung der verarbeiteten Stördaten Xo verarbeitet, um die verarbeiteten Daten D2 zu erzeugen, was ein Ergebnis der Eingabedatenverarbeitung an den Eingabedaten D1 ist.

[0019] Wenn es notwendig ist, unterschiedliche Datenumwandlungsverfahren zu verwenden, kann es erforderlich sein, mehrere Datenumwandlungen zu verbinden. In solch einem Fall wird eine Kombination aus einem Datenumwandlungsprozess, einer Verarbeitung umgewandelter Daten, einem Stördatenprozess, und einem Datenrückwandlungsprozess verwendet, und diese zahlreichen Datenumwandlungen werden so verbunden, dass ursprüngliche Daten nicht verarbeitet werden.

[0020] Nach dieser Erfindung ist es möglich, die Informationen während der Ausführung eines Verschlüsselungsalgorithmus zu verschleiern, die in dem Permutationsprozess und dem Substitutionsprozess zum Ersetzen von Daten und in dem Zugriffsprozess auf Datentabellen gewonnen werden können. Der Umwandlungsprozess, der gewährleistet, dass die korrekten Daten erhalten werden, ist eines der wirkungsvollen Verfahren, die für die Verschlüsselung und Entschlüsselung von Daten verwendet werden. In diesem Umwandlungsprozess wird das logische Exklusiv-ODER verwendet, um Daten in einem Datentauschprozess umzuwandeln, und die umgewandelten Daten und Stördaten werden durch dasselbe Verfahren in dem Datenprozess verarbeitet.

[0021] Eine typische Struktur der Erfindung ist wie folgt. Eine Informationsverarbeitungseinrichtung, umfassend: einen Speicher mit einer Programmspeichereinheit zum Speichern eines Programms und einer Datenspeichereinheit zum Speichern von Daten; eine Zentralverarbeitungseinheit zum Ausführen eines Datenprozesses in Übereinstimmung mit dem Programm, wobei das Programm eine oder mehrere Datenverarbeitungseinrichtungen umfasst, die je-

weils ein Prozessbefehl zum Erteilen eines Ausführungsbefehls an die Zentralverarbeitungseinheit sind; und eine Datenverarbeitungseinrichtung, wobei eine Datenverarbeitungseinrichtung Eingabedaten verarbeitet und die verarbeiteten Daten ausgibt, umfassend: eine Datenumwandlungsverarbeitungseinrichtung zum Umwandeln von Eingabedaten D1 unter Verwendung von Stördaten Xi, um umgewandelte Daten H1 zu erzeugen; eine Verarbeitungseinrichtung umgewandelter Daten zum Ausführen eines Operationsprozesses OP1 an den umgewandelten Daten H1 anstelle des Operationsprozesses OP1 an den Eingabedaten D1, der durch die Eingabedaten-Verarbeitungseinrichtung ausgeführt werden soll, um verarbeitete und umgewandelte Daten H2 zu erzeugen; eine Stördaten-Verarbeitungseinrichtung zum Ausführen des Operationsprozesses OP1 an den Stördaten Xi, um verarbeitete Stördaten Xo zu erzeugen; und eine Datenrückwandlungsverarbeitungseinrichtung zum Ausführen eines Operationsprozesses OP2 an den verarbeiteten und umgewandelten Daten H2 unter Verwendung der verarbeiteten Stördaten Xo zur Erzeugung verarbeiteter Daten D2, als ein Ergebnis des Operationsprozesses OP1 an den Eingabedaten D1.

[0022] Der Operationsprozess OP1 entspricht, zum Beispiel, dem Prozess einer Ausführungsform, die in **Fig. 4** dargestellt ist und später beschrieben wird. Der Operationsprozess OP1' entspricht, zum Beispiel, dem Verarbeiten von Stördaten 2 (**510** bis **513**, und **516** bis **520**).

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0023] **Fig. 1** ist ein Diagramm, das ein Beispiel der Hardware-Konstruktion einer bekannten IC-Karte darstellt.

[0024] **Fig. 2** ist ein Diagramm, das ein Beispiel der Hardware-Konstruktion eines bekannten IC-Kartenchips darstellt.

[0025] **Fig. 3** ist ein Diagramm, das Beispiele von Schwingungsformen des Verbrauchstroms einer IC-Karte darstellt.

[0026] **Fig. 4** ist ein Diagramm, das eine Vorgehensweise zur Datenumwandlung unter Verwendung eines Stördatensatzes nach einer Ausführungsform der Erfindung darstellt.

[0027] **Fig. 5** ist ein Diagramm, das eine Vorgehensweise zur Datenumwandlung unter Verwendung von zwei Stördatensätzen in einem Verschachtelungszustand nach einer Ausführungsform der Erfindung darstellt.

[0028] **Fig. 6** ist ein Diagramm, das eine Vorgehensweise zur Datenumwandlung unter Verwendung von zwei Stördatensätzen in einem kontinuierlichen Zustand nach einer Ausführungsform der Erfindung darstellt.

[0029] **Fig. 7** ist ein Diagramm, das eine Vorgehensweise zur Datenumwandlung nach einer Ausführungsform der Erfindung darstellt, wobei die Stör-

daten im Voraus verarbeitet werden.

[0030] **Fig. 8** ist ein Diagramm, das eine Vorgehensweise zur Datenumwandlung nach einer Ausführungsform der Erfindung darstellt, wobei die Rückwandlungsprozesse an den zwei Stördatensätzen vereinheitlicht sind.

[0031] **Fig. 9** ist ein Diagramm, das den gesamten Prozessablauf von DES nach einer Ausführungsform der Erfindung darstellt.

[0032] **Fig. 10** ist ein Diagramm, das den f Funktionsprozess von DES nach einer Ausführungsform der Erfindung darstellt.

[0033] **Fig. 11** ist ein Diagramm, das einen Umwandlungsprozess 1 nach einer Ausführungsform der Erfindung darstellt.

[0034] **Fig. 12** ist ein Diagramm, das einen IP Prozess nach einer Ausführungsform der Erfindung darstellt.

[0035] **Fig. 13** ist ein Diagramm, das einen PC-1 Prozess nach einer Ausführungsform der Erfindung darstellt.

[0036] **Fig. 14** ist ein Diagramm, das einen PC-2 Prozess nach einer Ausführungsform der Erfindung darstellt.

[0037] **Fig. 15** ist ein Diagramm, das einen LS Prozess nach einer Ausführungsform der Erfindung darstellt.

[0038] **Fig. 16** ist ein Diagramm, das einen auswählbaren Permutation E Prozess nach einer Ausführungsform der Erfindung darstellt.

[0039] **Fig. 17** ist ein Diagramm, das einen XOR Prozess zwischen einem Ergebnis des auswählbaren Permutation E Prozesses und einem Verschlüsselungscode nach einer Ausführungsform der Erfindung darstellt.

[0040] **Fig. 18** ist ein Diagramm, das ein S Box-Verfahren nach einer Ausführungsform der Erfindung darstellt.

[0041] **Fig. 19** ist ein Diagramm, das einen Permutation P Prozess nach einer Ausführungsform der Erfindung darstellt.

[0042] **Fig. 20** ist ein Diagramm, das einen XOR Prozess zwischen einem Ergebnis des Permutation P Prozesses und einem Ergebnis in der vorherigen Stufe darstellt.

[0043] **Fig. 21** ist ein Diagramm, das einen IP-1 Prozess nach einer Ausführungsform der Erfindung darstellt.

[0044] **Fig. 22** ist ein Diagramm, das einen Rückwandlungsprozess nach einer Ausführungsform der Erfindung darstellt.

[0045] **Fig. 23** ist ein Diagramm, das einen Prozess zur Ausbildung einer umgewandelten S Box-Tabelle nach einer Ausführungsform der Erfindung darstellt.

[0046] **Fig. 24** ist ein Diagramm, das eine Ausbildungsroutine einer i-ten umgewandelten S Box-Tabelle nach einer Ausführungsform der Erfindung darstellt.

[0047] **Fig. 25** ist eine i-te S Box-Tabelle nach einer Ausführungsform der Erfindung.

[0048] **Fig. 26** ist eine Tabelle, die umgewandelte Daten von der i-ten S Box-Tabelle der Ausführungsform aufweist.

[0049] **Fig. 27** ist eine Tabelle, die umgewandelte Positionen der i-ten S Box-Tabelle der Ausführungsform aufweist.

[0050] **Fig. 28** ist eine auswählbare Permutation E Tabelle nach einer Ausführungsform der Erfindung.

[0051] **Fig. 29** ist eine Permutation P Tabelle nach einer Ausführungsform der Erfindung

[0052] **Fig. 30** ist ein Diagramm, das einen Verschlüsselungsprozess an Stördaten nach einer Ausführungsform der Erfindung darstellt.

[0053] **Fig. 31** ist ein Diagramm, das einen Umwandlungsrechenprozess an Verschlüsselungsdaten für Stördaten nach einer Ausführungsform der Erfindung darstellt.

[0054] **Fig. 32** ist ein Diagramm, das einen Entschlüsselungsprozess an Stördaten nach einer Ausführungsform der Erfindung darstellt.

[0055] **Fig. 33** zeigt ein Beispiel einer ursprünglichen Tabelle nach einer Ausführungsform der Erfindung.

[0056] **Fig. 34** zeigt eine Tabelle, deren Inhalte in Bezug auf diejenigen der Tabelle, die in **Fig. 33** gezeigt ist, nach einer Ausführungsform der Erfindung modifiziert sind.

[0057] **Fig. 35** zeigt eine Tabelle, deren Anordnung in Bezug auf diejenige der Tabelle, die in **Fig. 34** gezeigt ist, nach einer Ausführungsform der Erfindung modifiziert ist.

[0058] **Fig. 36** ist ein Diagramm, das einen Umwandlungsprozess b nach einer Ausführungsform der Erfindung darstellt.

[0059] **Fig. 37** ist eine IP Permutationstabelle nach einer Ausführungsform der Erfindung.

[0060] **Fig. 38** ist eine PC-1 auswählbare Permutationstabelle nach einer Ausführungsform der Erfindung

AUSFÜHRLICHE BESCHREIBUNG DER AUSFÜHRUNGSFORMEN

[0061] Die Ausführungsformen der Erfindung werden mit Bezugnahme auf die beigefügten Zeichnungen beschrieben.

[0062] **Fig. 1** ist eine einfache Ansicht einer IC-Karte. Die Position eines IC-Kartenchips **102** und die Anzahl der Kontakte und ihre Operationszuweisungen auf der IC-Karte **101** sind in ISO 7816 durch die Technischen Daten definiert.

[0063] **Fig. 2** zeigt den inneren Aufbau eines IC-Kartenchips **102**. Der Aufbau ist schon im Zusammenhang mit herkömmlichen Technologien beschrieben worden. Nach dieser Erfindung werden Daten, die durch das Programm **205** verarbeitet werden sollen, so gestört, dass es schwierig wird, die ursprünglichen Daten von der Schwingungsform des Stroms, der durch die Hardware des IC-Kartenchips während der Datenverarbeitung verbraucht wird, abzuleiten.

Das Grundkonzept wird beschrieben, indem die folgende einfache Befehlsfolge als Beispiel dient:
logica_shift1 R1

(2)
xor R1 R2

(3)
[0064] Die Gleichung (2) ist ein Befehl für logischen Schleifendurchlauf des Wertes in einem Register R1 nach links. Das höchstwertige Bit bewegt sich zu dem niedrigstwertigen Bit. Das Ergebnis wird in das Register R1 geladen. Das logische Exklusiv-ODER zwischen dem Ergebnis und dem Wert in einem Register R2 wird durch die Gleichung (2) berechnet, und dieses Ergebnis wird in das Register R2 geladen. Diese Operationen werden durch diese Befehlsfolge durchgeführt. Solche Befehle werden häufig in einem Verschlüsselungsalgorithmus, wie DES, verwendet. Weil in den Gleichungen (2) und (3) die Prozessdaten selbst verwendet werden, verändert sich die Amplitude des Stromverbrauchs mit den Inhalten der Prozessdaten. Demzufolge besteht eine Möglichkeit, die Daten durch Überwachen der Stromverbrauchsform abzuleiten.

[0065] Damit in den Gleichungen (2) und (3) die Prozessdaten selbst nicht verwendet werden, wird die Befehlsfolge geändert auf:
xor X1 R1

(4)
xor X2 R2

(5)
logica_shift1 R1

(6)
xor R1 R2

(7)
logica_shift1 X1

(8)
xor X1 X2

(9)
xor X2 R2

(10)
wobei X1 und X2 beliebige Zufallszahlen und Daten für Störung sind. Mit den Gleichungen (4) und (5) wird ein logisches Exklusiv-ODER zwischen R1 und X1 und zwischen R2 und X2 zum Ausführen eines Umwandlungsprozesses zum Umwandeln ursprünglicher Daten berechnet. Obwohl die Gleichungen (6) und (7) genauso ausgedrückt sind wie die Gleichungen (2) und (3), sind die Werte R1 und R1 in den Gleichungen (6) und (7) nicht die Werte von ursprünglichen Daten, weil der Umwandlungsprozess ausgeführt wurde. Mit den Gleichungen (8) und (9) werden

die Stördaten selbst verarbeitet. Mit der Gleichung (10) wird ein logisches Exklusiv-ODER zwischen den verarbeiteten Stördaten und dem Verarbeitungsergebnis der Gleichung (7) berechnet, um einen Rückwandlungsprozess zum Wiederherstellen der ursprünglichen Daten auszuführen.

[0066] Diese Prozesse werden insbesondere unter Verwendung von bestimmten numerischen Werten beschrieben. R1 und R2 haben die folgenden Werte:
R1:11001010

(11)
R2:01010111

(12)
[0067] Der Wert von R1, der durch die Gleichung (2) verarbeitet wurde ist:
R1:10010101

(13)
[0068] Das Verarbeitungsergebnis durch die Gleichung (3) ist:
R2:11000010

(14)
[0069] Eine Modifizierung dieser Erfindung wird beschrieben. Zuerst haben die Stördaten die folgenden Werte, wobei dieselben Werte von R1 und R2 verwendet werden:
X1:10010111

(15)
X2:00111010

(16)
[0070] Die Verarbeitungsergebnisse durch die Gleichungen (4) und (5) sind:
R1:01011101

(17)
R2:01101101

(18)
[0071] Die Verarbeitungsergebnisse durch die Gleichungen (6) und (7) sind:
R1:10111010

(19)
R2:11010111

(20)
[0072] Die Verarbeitungsergebnisse der Daten X1 und X2 zur Störung durch die Gleichungen (8) und (9) sind:
X1:00101111

(21)
X2:00010101

(22)

[0073] Das Ergebnis des Rückwandlungsprozesses durch die Gleichung (10) ist dasselbe wie das Ergebnis des ursprünglichen Datenprozesses, das durch die Gleichung (14) angezeigt ist:
R2:11000010

(23)

[0074] Wie anhand dieses Beispiels gezeigt, werden die umgewandelten ursprünglichen Daten und die Stördaten in einer ähnlichen Weise verarbeitet, und die Ergebnisse werden zurückgewandelt, um die ursprünglichen Daten wiederherzustellen. Weil in diesem Datenprozess nicht die ursprünglichen Daten selbst verwendet werden, ist es schwierig, die ursprünglichen Daten von der Stromschwingungsform abzuleiten, obwohl die umgewandelten Daten abgeleitet werden können.

[0075] Das obige Beispiel, in dem bestimmte numerische Werte verwendet werden, wird nun durch ein allgemeines Format ausgedrückt. Der tatsächliche Prozess ist wie folgt:

$\text{Output}(j) = f(\text{Input}(i))$

(24)

[0076] Dieser Prozess bedeutet, dass i Eingaben einem Prozess f unterzogen werden, um j Ausgaben auszugeben. In dem durch die Gleichungen (2) und (3) gezeigten Beispiel gibt es zwei Eingaben $R1$ und $R2$ und eine Ausgabe, die in dem Register $R2$ gespeichert ist. Damit es schwierig wird, die ursprünglichen Daten von der Stromschwingungsform während des Verarbeitens der Gleichung (24) abzuleiten, werden die folgenden Gleichungen verwendet:

$\text{InputX}(i) = h(\text{Input}(i), X(i))$

(25)

$\text{OutputX}(j) = f(\text{InputX}(i))$

(26)

$X_{\text{output}}(j) = f(X(i))$

(27)

$\text{Output}(j) = g(\text{OutputX}(i), X_{\text{output}}(i))$

(28)

[0077] Die Gleichung (25) wandelt die Eingabedaten $\text{Input}(i)$ unter Verwendung von Stördaten $X(i)$ um, um die umgewandelten Eingabedaten $\text{InputX}(i)$ zu erzeugen. Diese Umwandlungsoperation ist durch h repräsentiert. Die Gleichung (26) ist ein Datenprozess zum Verarbeiten von Daten unter Verwendung von umgewandelten Eingabedaten. Die Gleichung (27) ist ein Stördatenprozess zum Verarbeiten der Stördaten in einer ähnlichen Weise wie die Eingabedaten. Die Gleichung (28) ist ein Rückwandlungsprozess zum umgekehrten Verarbeiten des Verarbeitungsergebnisses umgewandelter Eingabedaten $\text{OutputX}(j)$ und des Verarbeitungsergebnisses von Stördaten $X_{\text{output}}(j)$. Die Rückwandlungsoperation wird durch g repräsentiert.

[0078] Das Verarbeiten durch die Gleichung (25) entspricht den Gleichungen (4) und (5) in dem oben beschriebenen Beispiel, und die Umwandlungsoperation h entspricht dem logischen Exklusiv-ODER. Die Verarbeitung umgewandelter Eingabedaten durch die Gleichung (26) entspricht den Gleichungen (6) und (7). Die Gleichungen (8) und (9) zeigen das Verarbeiten von Daten für die Stördaten in der Gleichung (27). Der Rückwandlungsprozess durch die Gleichung (28) entspricht der Gleichung (10). Die Rückwandlungsoperation g entspricht dem logischen Exklusiv-ODER.

[0079] Welche Operationen für die Umwandlungsoperation h und die Rückwandlungsoperation g ausgewählt werden, wird durch die Charakteristika des Datenprozesses f bestimmt. Bei der Verarbeitung durch die Gleichungen (2) und (3) ist das logische Exklusiv-ODER die Umwandlungsoperation h und ebenfalls die Rückwandlungsoperation (umgekehrte) Operation g . Für eine Verschiebeoperation und eine XOR Operation, durch Auswählen des logischen Exklusiv-ODER als die Umwandlungsoperation h , ist die Rückwandlungsoperation (umgekehrte) Operation g das logische Exklusiv-ODER. Dieses ist so, weil das logische Exklusiv-ODER zwischen denselben Daten logische 0 ist und die Operation des logischen Exklusiv-ODER verschwunden ist.

[0080] Wenn der Datenprozess f Addition/Subtraktion ist, kann Addition oder Subtraktion als die Umwandlungsoperation h ausgewählt werden und die entsprechende umgekehrte Operation g ist Subtraktion oder Addition. Zum Beispiel kann die folgende Operation:

$\text{Output} = \text{Input}(1) + \text{Input}(2) - \text{Input}(3)$

(29)

umgewandelt werden in:

$\text{InputX}(1) = \text{Input}(1) + X(1)$

(30)

$\text{InputX}(2) = \text{Input}(2) + X(2)$

(31)

$\text{InputX}(3) = \text{Input}(3) + X(3)$

(32)

[0081] Durch Verarbeiten der umgewandelten Eingabedaten, kann das Verarbeitungsergebnis der umgewandelten Eingabedaten erhalten werden:

$\text{OutputX} = \text{InputX}(1) + \text{InputX}(2) - \text{InputX}(3)$

(33)

[0082] Die Stördaten werden auf ähnliche Weise verarbeitet:

$X_{\text{output}} = X(1) + X(2) - X(3)$

(34)

[0083] Als Nächstes wird der Rückwandlungsprozess ausgeführt:

$\text{Output} = g(X_{\text{output}}) = \text{OutputX} - X_{\text{output}} = \text{InputX}(1)$

+ InputX(2) – InputX(3) – (X(1) + X(2) – X(3)) = Input(1) + Input(2) – Input(3)

(35)

[0084] Auf die obige Weise können die ursprünglichen Daten erhalten werden. Dieses ist so, weil die ursprünglichen Daten bei der Additions-/Subtraktionsrechnung durch Addieren eines bestimmten Wertes und Subtrahieren des addierten Wertes von dem letzten Verarbeitungsergebnis erhalten werden können.

[0085] Für den Datenprozess f von Multiplikation/Division kann der Umwandlungsprozess und der Rückwandlungsprozess durch Auswählen von Multiplikation oder Division als Umwandlungsoperation h und Division oder Multiplikation als Rückwandlungsoperation durchgeführt werden. Dieses ist so, weil, ähnlich der Addition/Subtraktion, die ursprünglichen Daten in der Multiplikations-/Divisionsrechnung durch Multiplizieren mit (Dividieren durch) einem (einen) bestimmten Wert und Dividieren (Multiplizieren) des letzten Verarbeitungsergebnisses durch den (mit dem) Wert erhalten werden können.

[0086] Für den Datenprozess f von Addition und Subtraktion in Modularrechnung kann Addition und Subtraktion der Zahl, die den Modulus N mit einer beliebigen ganzen Zahl multipliziert, als die Umwandlungsoperation h ausgewählt werden. Betrachtet man zum Beispiel die folgende Addition und Subtraktion in Modularrechnung:

$$\text{Output} = (\text{Input}(1) + \text{Input}(2) - \text{Input}(3)) \bmod N$$

(36)

[0087] Die Eingabedaten Input(i) werden umgewandelt:

$$\text{InputX}(i) = \text{Input}(i) + k(i) * N$$

(37)

[0088] Unter Verwendung der umgewandelten Eingabedaten wird die Addition und Subtraktion in Modularrechnung ausgeführt:

$$\text{OutputX} = (\text{InputX}(1) + \text{InputX}(2) - \text{InputX}(3)) \bmod N$$

(38)

[0089] Die Gleichung (38) kann unter Verwendung der Gleichung (37) umgewandelt werden:

$$\text{OutputX} = ((\text{Input}(1) + k(1) * N) + (\text{Input}(2) + k(2) * N) - (\text{Input}(3) + k(3) * N)) \bmod N = ((\text{Input}(1) + \text{Input}(2) - \text{Input}(3)) + (k(1) * N + k(2) * N - k(3) * N)) \bmod N$$

(39)

[0090] Unter Verwendung der Eigenschaft der Modularrechnung:

$$0 = k * N \bmod N$$

(40)

wird der Wert in den zweiten Klammern der Gleichung (39) 0 und die Gleichung (39) ist gegeben durch:

$$\text{OutputX} = (\text{Input}(1) + \text{Input}(2) - \text{Input}(3)) \bmod N$$

(41)

[0091] Das Rechenergebnis der umgewandelten Eingabedaten ist nämlich dasselbe wie das Rechenergebnis der ursprünglichen Daten. Dieses ist ein Beispiel dafür, dass der Stördatenprozess und der Rückwandlungsprozess unnötig sind, wenn das oben beschriebene Merkmal der Modularrechnung einbezogen wird. Das Verarbeitungsergebnis von Stördaten:

$$\text{Xoutput}(i) = (k(1) * N + k(2) * N - k(3) * N) \bmod N$$

(42)

ist 0, so dass der Stördatenprozess und der Rückwandlungsprozess unnötig sind.

[0092] Für den Datenprozess f von Multiplikation in Modularrechnung kann ein ganzzahliges Vielfaches des Modulus N addiert mit 1 für die Umwandlungsoperation h verwendet werden. Man betrachte zum Beispiel die folgende Multiplikation in Modularrechnung:

$$\text{Output} = \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \bmod N$$

(43)

[0093] Die Eingabedaten Input(i) werden umgewandelt:

$$\text{InputX}(i) = \text{Input}(i) * (k(i) * N + 1)$$

(44)

[0094] Unter Verwendung der umgewandelten Eingabedaten wird die Multiplikations-Modularrechnung ausgeführt:

$$\text{OutputX} = \text{InputX}(1) * \text{InputX}(2) * \text{InputX}(3) \bmod N$$

(45)

[0095] Die Gleichung (45) kann unter Verwendung der Gleichung (44) umgewandelt werden:

$$\text{OutputX} = (\text{Input}(1) * (k(1) * N + 1)) * (\text{Input}(2) * (k(2) * N + 1)) * (\text{Input}(3) * (k(3) * N + 1)) \bmod N = (\text{Input}(1) * \text{Input}(2) * \text{Input}(3)) * ((k(1) * N + 1) * (k(2) * N + 1)) * ((k(3) * N + 1)) \bmod N$$

(46)

[0096] Unter Verwendung der Eigenschaft von Modularrechnung:

$$0 = k * N \bmod N$$

(47)

ist die Gleichung (46) gegeben durch:

$$\text{OutputX} = (\text{Input}(1) * \text{Input}(2) * \text{Input}(3)) * (1 * 1 * 1) \bmod N = \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \bmod N$$

(48)

[0097] Das Rechenergebnis der umgewandelten Eingabedaten ist nämlich dasselbe wie das Rechenergebnis der ursprünglichen Daten. Dieses ist ebenfalls ein Beispiel dafür, dass die Datenverarbeitung der Stördaten und die Rückwandlungsoperation aufgrund der Eigenschaft von Modularrechnung unnötig sind.

[0098] Für die Funktion f von Multiplikation mit einer ganzen Zahl in Modularrechnung mit Modulus N kann die Umwandlungsoperation h als Multiplikation mit einer reversiblen Zahl x in dieser Modularrechnung ausgewählt werden. In diesem Fall ist g die Multiplikation mit Y , so dass:

$$1 = X * Y \bmod N$$

(49)

[0099] Ein einfaches Beispiel von solch numerischen Werten ist, dass $X = 2$ und $Y = (N + 1)/2$; (N ist eine ungerade Zahl). Das Verarbeitungsergebnis von ursprünglichen Daten kann durch Multiplizieren von X als der Umwandlungsoperation und Multiplizieren von Y in dem Rückwandlungsprozess mit der Anzahl von X -Multiplikationen erhalten werden. Man betrachte zum Beispiel die folgende Addition und Subtraktion in Modularrechnung:

$$\text{Output} = \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \bmod N$$

(50)

[0100] Die Eingabedaten $\text{Input}(i)$ werden umgewandelt:

$$\text{InputX}(i) = \text{Input}(1) * X$$

(51)

[0101] Unter Verwendung der umgewandelten Eingabedaten wird die Additions-/Subtraktions-Modularrechnung durchgeführt:

$$\text{OutputX} = \text{InputX}(1) * \text{InputX}(2) * \text{InputX}(3) \bmod N$$

(52)

[0102] Wenn ein Wert mit Y multipliziert wird, so oft X multipliziert wird, für den Rückwandlungsprozess g ausgewählt wird, können die ursprünglichen Daten unter Durchführung des Rückwandlungsprozesses g für den outputX erhalten werden:

$$\begin{aligned} \text{Output} &= \text{OutputX} * Y * Y * Y \bmod N = \text{InputX}(1) * \text{InputX}(2) * \text{InputX}(3) * Y * Y * Y \bmod N = \text{Input}(1) * X * \text{Input}(2) * X * \text{Input}(3) * X * Y * Y * Y \bmod N = \text{Input}(1) * \text{Input}(2) * \text{Input}(3) * X * X * X * Y * Y * Y \bmod N = \\ &= \text{Input}(1) * \text{Input}(2) * \text{Input}(3) * X * Y * X * Y * X * Y * X * Y \bmod N = \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \bmod N \end{aligned}$$

(53)

[0103] Für die Gleichung (53) wurde das Merkmal der Gleichung (49) verwendet. In diesem Beispiel kann, obwohl der Stördatenprozess unnötig ist, das korrekte Ergebnis, in dem Rückwandlungsprozess, erhalten werden, wenn die Umkehrung in dem Modulus N mit dem Wert der Anzahl der Stördaten-Multiplikationen multipliziert wird.

[0104] Es ist ebenfalls notwendig, eine Wiedergewinnungsoperation von Daten aus einer Tabelle zu stören, damit Daten nicht von der Stromschwingungsform abgeleitet werden können. Ein Beispiel zur Störung von Tabellendaten und Störung von Tabellenadressen wird beschrieben, indem das Wiedergewinnen von Daten aus einer Tabelle, die in **Fig. 33** gezeigt ist, als ein Beispiel genommen wird.

[0105] Ein logisches Exklusiv-ODER zwischen den Tabellendaten, die in **Fig. 33** gezeigt sind, und Stördaten $X1$ wird berechnet. Zum Beispiel werden die Stördaten von „9“ ausgewählt und ein logisches Exklusiv-ODER zwischen dem Tabellenwert und „9“ wird berechnet. Die Ergebnisse werden in der Tabelle von **Fig. 34** dargestellt. Als Nächstes wird, um die Tabellenadresse zu stören, ein logisches Exklusiv-ODER zwischen der Reihenzahl und den ausgewählten Stördaten $X2$ von „3“ und zwischen der Spaltenzahl und ausgewählten Stördaten $X3$ von „2“ berechnet, um die Tabelle umzuordnen. Die Ergebnisse werden in **Fig. 35** gezeigt. Die Daten **3301** von „0“ in der ersten Reihe und zweiten Spalte der ursprünglichen Tabelle, die in **Fig. 33** gezeigt ist, sind auf die Daten **3401** von „9“ in der Tabelle, die in **Fig. 34** gezeigt ist, geändert, nachdem das logische Exklusiv-ODER der Stördaten $X1$ berechnet ist. Nachdem die logischen Exklusiv-ODER zwischen der Reihenzahl und Stördaten $X2$ und zwischen der Spaltenzahl und Stördaten $X3$ berechnet sind, bewegen sich die Daten **3401** zu der Position von Daten **3501**. Derartige Tabellen werden hergestellt, um die Wiedergewinnungsoperation von Daten aus der Tabelle zu stören. [0106] Es wird hierbei angenommen, dass eine Reihenzahlvariable $Gyou$ und eine Spaltenzahlvariable $Retsu$ schon durch ein logisches Exklusiv-ODER von Stördaten $Y1$ und $Y2$ umgewandelt worden sind, bevor die Adressenrechnung ausgeführt wird. Es wird nämlich angenommen, dass korrekte Reihen- und Spaltenzahlen $Gyou$ und $Retsu$ erst erhalten werden können, wenn das logische Exklusiv-ODER zwischen den $Y1$ und $Y2$ und den $Gyou$ und $Retsu$ berechnet ist. Dieses kann durch die folgenden Beziehungen ausgedrückt werden:

$$Gyou = GyouY1 \text{ xor } Y1$$

(54)

$$Retsu = RetsuY2 \text{ xor } Y2$$

(55)

[0107] Wenn jedoch die Tabelle, die in **Fig. 33** gezeigt ist, für diesen Rückwandlungsprozess verwendet wird, werden die korrekten Adressendaten verwendet, so dass die Adressendaten von der Stromschwingungsform abgeleitet werden können. Demzufolge werden, zuerst, die Stördaten $X2$ und $X3$, die zum Stören der Reihen- und Spaltenzahlen verwendet wurden, als die Tabelle, die in **Fig. 35** gezeigt ist, gebildet wurde, verwendet:

$$GyouY1X2 = GyouY1 \text{ xor } X2$$

(56)

$$RetsuY1X3 = RetsuY2 \text{ xor } X3$$

(57)

[0108] Die verwendeten Stördaten werden dann dem Rückwandlungsprozess unterzogen:

$$GyouX2 = GyouY1X2 \text{ xor } Y1$$

(58)

$$\text{RetsuX3} = \text{RetsuY2X3} \text{ xor } \text{Y2}$$

(59)

[0109] Bei dieser Vorgehensweise werden die korrekten Reihen- und Spaltenzahlen nicht verwendet, so dass es schwierig ist, die korrekten Reihen- und Spaltenzahlen von der Stromschwingungsform abzuleiten. Unter Verwendung von GyouX2 und RetsuX3 und Verweisung auf die in **Fig. 35** gezeigte Tabelle TableX1X2X3, werden Daten X1 erhalten:

$$\text{DataX1} = \text{TableX1X2X3}(\text{GyouX2}, \text{RetsuX3})$$

(60)

[0110] Weil die in **Fig. 35** gezeigte Tabelle schon durch die Stördaten X1 umgewandelt ist, werden die Stördaten X1 in dem nachfolgenden Prozess verwendet. In den Prozessen der Gleichungen (56) bis (60) werden keine ursprünglichen Daten verwendet.

[0111] Jedes Mal, wenn eine vorgegebene Anzahl an Prozessen ausgeführt ist, werden die Tabellendaten-Stördaten X1 und Reihen- und Spaltenzahlen-Stördaten X2 und X3 zufällig erzeugt, um die Tabellen umzuwandeln. Die Tabellen werden mit diesem Umwandlungsprozess so umgewandelt, dass es schwierig ist, die Daten von der Stromschwingungsform abzuleiten.

[0112] In dem Vorhergehenden sind die Art von Stördaten und ein Verfahren zum Umwandeln von Daten beschrieben worden. Als Nächstes werden die sequentiellen Verarbeitungen dafür beschrieben werden. In **Fig. 4** ist eine Ausführungsform einer grundsätzlichen Vorgehensweise zur Informationsverschleierung unter Verwendung von Stördaten dargestellt.

[0113] In **Fig. 4** ist eine grundsätzliche Vorgehensweise dargestellt. Eine Stördaten-Erzeugungseinheit erzeugt Stördaten Xi (**401**). Dafür gibt es, als ein allgemeines Verfahren, ein Verfahren zur Erzeugung einer Zufallszahl mit einer notwendigen Länge unter Verwendung eines Generators von Zufallszahlen oder eines Generators pseudozufälliger Zahlen. Als Nächstes wandelt eine Datenumwandlungsverarbeitungseinheit (**406**) Eingabedaten D1 (**405**) durch die Stördaten Xi zur Erzeugung von umgewandelten Daten H1 (**407**) um. Wie schon vorher beschrieben, kann der Umwandlungsprozess ein logisches Exklusiv-ODER, Addition und Subtraktion, Multiplikation und Division, oder dergleichen sein. Eine Verarbeitungseinheit umgewandelter Daten verarbeitet die umgewandelten Daten H1 (**408**) zur Erzeugung verarbeiteter und umgewandelter Daten H2. Eine Stördaten-Verarbeitungseinheit führt (**403**) einen dem der Eingabedaten ähnlichen Datenprozess zur Erzeugung verarbeiteter Stördaten Xo (**404**) durch. Eine Datenrückwandlungsverarbeitungseinheit erzielt (**410**) korrekte, verarbeitete Daten D2 unter Verwendung der verarbeiteten Stördaten Xo und verarbeiteten und umgewandelten Daten D2 (**411**). Der Prozess, der durch die Datenumwandlungseinheit (**406**)

und die Datenrückwandlungseinheit ausgeführt werden soll, kann, wie vorher beschrieben, ein logisches Exklusiv-ODER, Addition und Subtraktion, Multiplikation und Division, Modularrechnung, oder dergleichen umfassen.

[0114] In der in **Fig. 5** dargestellten Ausführungsform werden zwei Sätze von Stördaten verwendet. Eine Vorgehensweise zur Informationsverschleierung, bei der erste Stördaten verwendet werden, enthält eine weitere Vorgehensweise zur Informationsverschleierung, bei der zweite Stördaten verwendet werden. Der Hauptablauf ist der Ausführungsform, die in **Fig. 4** gezeigt ist, ähnlich. Umgewandelte Daten H1 (**507**), die durch die ersten Stördaten umgewandelt wurden, werden verarbeitet. Die verarbeiteten und umgewandelten Daten H2 (**509**) werden durch eine zweite Datenumwandlungsverarbeitungseinheit (**510**) unter Verwendung der zweiten Stördaten X2i umgewandelt, um dadurch verarbeitete und umgewandelte Daten H3 (**511**) zu erzeugen. Diese verarbeiteten und umgewandelten Daten H3 werden durch eine zweite Verarbeitungseinheit umgewandelter Daten (**512**) zur Erzeugung von verarbeiteten und umgewandelten Daten H4 verarbeitet. Eine zweite Datenrückwandlungsverarbeitungseinheit (**520**) wandelt die zweiten Stördaten zurück zur Erzeugung von verarbeiteten und umgewandelten Daten H5 (**521**). Eine erste Datenrückwandlungsverarbeitungseinheit (**514**) wandelt die ersten Stördaten zurück, um korrekte, verarbeitete Daten D2 (**515**) zu erhalten. Ein Beispiel der Umwandlung unter Verwendung eines logischen Exklusiv-ODER ist wie folgt:

$$\text{H1} = \text{D1} \text{ xor } \text{X1i}$$

$$\text{H2} = \text{f1}(\text{H1})$$

$$\text{X1o} = \text{f1}(\text{X1i})$$

$$\text{H31} = \text{H2} \text{ xor } \text{X2i}$$

$$\text{H32} = \text{D2} \text{ xor } \text{X2i}$$

$$\text{H4} = \text{f2}(\text{H31}, \text{H32})$$

$$\text{X2o} = \text{f2}(\text{X2i}, \text{X2i})$$

$$\text{H5} = \text{H4} \text{ xor } \text{X2o}$$

$$\text{D2} = \text{H5} \text{ xor } \text{X1o}$$

(61)

[0115] Die Datenprozesse sind durch f1 und f2 dargestellt. Weil in diesem Beispiel in dem zweiten Datenprozess f2 andere Daten D2 verwendet werden und diese Daten D2 durch die zweiten Stördaten umgewandelt werden, ist die Vorgehensweise dieser Ausführungsform wirkungsvoll.

[0116] In der in **Fig. 6** dargestellten Ausführungsform werden ebenfalls zwei Sätze von Stördaten verwendet. Ein bedeutender Unterschied zu der in **Fig. 5** dargestellten Ausführungsform besteht darin, dass eine Vorgehensweise zur Informationsverschleierung, bei der erste Stördaten verwendet werden, kontinuierlich mit einer weiteren Vorgehensweise zur Informationsverschleierung, bei der zweite Stördaten verwendet werden, abläuft. Bei dieser Vorgehensweise zur Verschleierung des Prozesses, der korrekte Daten verwendet, werden, vor dem Rückwand-

lungsprozess der ersten Stördaten, die Eingabedaten unter Verwendung der zweiten Stördaten weiter umgewandelt. Umgewandelte Daten H1 (**609**), die durch die ersten Stördaten umgewandelt wurden, werden durch eine erste Verarbeitungseinheit für umgewandelte Daten (**610**) verarbeitet. Die verarbeiteten und umgewandelten Daten H2 (**611**) werden durch eine zweite Erzeugungseinheit umgewandelter Daten (**612**) unter Verwendung der zweiten Stördaten H2 (**611**) umgewandelt, um verarbeitete und umgewandelte Daten H3 (**613**) zu erzeugen. Eine erste Datenrückwandlungsverarbeitungseinheit (**605**) wandelt für die ersten Stördaten zurück. Dieses Verarbeitungsergebnis von verarbeiteten und umgewandelten Daten H4 (**606**) wird für eine zweite Verarbeitungseinheit umgewandelter Daten (**614**) zur Erzeugung von verarbeiteten und umgewandelten Daten H5 (**615**) verwendet. Eine zweite Datenrückwandlungsverarbeitungseinheit (**616**) wandelt die Daten zur Erzeugung von korrekten, verarbeiteten Daten D2 (**617**) zurück. Ein Beispiel der Umwandlung unter Verwendung eines logischen Exklusiv-ODER ist wie folgt:

H1 = D1 xor X1o
H2 = f1(H1)
X1o = f1(X1i)
H3 = H2 xor X2i
H4 = H3 xor X1o
H5 = f2(H4)
X2o = f2(X2i)
D2 = H5 xor X2i

(62)

[0117] Diese Vorgehensweise ist in dem Fall wirkungsvoll, bei dem es eine Vielzahl an Prozessen gibt und eine Vielzahl an Sätzen von Stördaten verwendet werden.

[0118] In der in **Fig. 7** gezeigten Ausführungsform werden die Stördaten im Voraus verarbeitet, um die Vorgehensweise effizient zu gestalten. Eine Stördaten-Verarbeitungseinheit erzeugt im Voraus verarbeitete Stördaten X_o (**703**), welche in einer Speichereinheit für verarbeitete Stördaten (**706**) gespeichert werden. Während der Ausführung liest eine Datenrückwandlungsverarbeitungseinheit (**713**) die gespeicherten, verarbeiteten Stördaten (**714**), um sie zu verwenden. Diese Vorgehensweise ist effizient, wenn ähnliche Datenprozesse mehrmalig ausgeführt werden. Weil die Stördaten jedoch mehrmalig verwendet werden, ist es wirkungsvoller, die Stördaten jedes Mal zu ändern, wenn der Datenprozess ausgeführt wird, wie bei der in **Fig. 4** gezeigten Ausführungsform. Dieses kann durch Abwägen zwischen Verarbeitungszeit und Informationssicherheit festgelegt werden.

[0119] In der in **Fig. 8** gezeigten Ausführungsform ist der Rückwandlungsprozess für die ersten und zweiten Stördaten vereinheitlicht, und unter Verwendung des vereinheitlichten Ergebnisses werden die Daten danach zurückgewandelt. Erste und zweite

Stördaten-Verarbeitungseinheiten (**803** und **807**) verarbeiten die ersten und zweiten Stördaten zur Erzeugung von verarbeiteten Stördaten X_{1o} und X_{2o}. Diese Daten werden durch eine Datenrückwandlungs- und Vereinheitlichungseinheit zur Erzeugung von vereinheitlichten und verarbeiteten Stördaten X_o vereinheitlicht. Unter Verwendung dieser Daten wandelt eine Rückwandlungsverarbeitungseinheit (**820**) die verarbeiteten und umgewandelten Daten H4 (**819**), die durch erste und zweite Verarbeitungseinheiten für umgewandelte Daten (**814** und **818**) verarbeitet wurden, zur Erzeugung korrekter, verarbeiteter Daten D2, zurück. Mit dieser Vorgehensweise werden die verarbeiteten Stördaten vereinheitlicht und ein vereinheitlichter Rückwandlungsprozess wird danach ausgeführt, anstatt den Rückwandlungsprozess unabhängig auszuführen. Diese Vorgehensweise ist in dem Fall wirkungsvoll, wenn der Rückwandlungsprozess eine lange Verarbeitungszeit in Anspruch nimmt.

[0120] Als Nächstes werden Ausführungsformen beschrieben, bei denen symmetrische Verschlüsselung DES (Standard der Datenverschlüsselung) verwendet wird. Die Erfindung ist auf andere Verschlüsselungssysteme anwendbar.

[0121] Mit DES wird Verschlüsselung und Entschlüsselung von 64-Bit Daten (Klartext oder Verschlüsselungstext) unter Verwendung eines Verschlüsselungscodes von 56 Bits ausgeführt. Weil derselbe Verschlüsselungscode sowohl für Verschlüsselung als auch für Entschlüsselung verwendet wird, nennt man DES eine symmetrische Verschlüsselung. So wie Trumpfkarten zufällig umgedreht werden, werden Bits von Klartext (der verschlüsselt werden soll) zufallsweise ausgetauscht und verschlüsselt. Datentausch wird gemäß dem Verschlüsselungscode durchgeführt. Wenn Daten entschlüsselt werden, werden Bits von einem Verschlüsselungstext in der umgekehrten Reihenfolge der Verschlüsselung zur Wiederherstellung der ursprünglichen Daten ausgetauscht. Beim Datentausch von DES werden zwei Tauschverfahren verwendet, wobei das eine auf der Basis einer 1-Bit-Einheit und das andere auf der Basis einer Mehr-Bit-Einheit erfolgt. Das erstere wird Permutation genannt und das letztere wird Substitution genannt.

[0122] Mit Bezugnahme auf **Fig. 9** wird DES Verschlüsselung beschrieben. Ein Umwandlungsprozess a (**901**), ein Umwandlungsprozess b (**902**), und ein Rückwandlungsprozess (**916**) gehören zu der vorliegenden Erfindung und sind für die grundsätzliche Verschlüsselung von DES nicht relevant. Ein Verschlüsselungstext wird zuerst einer anfänglichen Permutation (IP) 902 unterzogen. Diese Permutation wird unter Verwendung einer Anfangspermutationstabelle zum Tauschen von 64-Bit Daten des Verschlüsselungstextes auf der Basis einer 1-Bit-Einheit durchgeführt. Eine Serie solcher Operationen wird in sechzehn Stufen wiederholt, um die Permutation (IP⁻¹) **915** der Anfangspermutation umzukehren.

[0123] Bei jeder Stufe wird ein Prozess berechnet, der eine f Funktion **903** genannt wird, indem Daten von 32 Bits von entweder der ersten oder der zweiten Hälfte in der vorhergehenden Stufe und der Verschlüsselungscode eingegeben werden, und dann wird unter Verwendung der Ausgabe der f Funktion und 32 Bits der verbleibenden Hälfte in der vorhergehenden Stufe eine logische Exklusiv-ODER Operation **909** durchgeführt. Daten des Verschlüsselungscode werden ebenfalls ausgetauscht. Daten des Verschlüsselungscode werden zuerst auswählbarer Permutation PC-1 (**905**) unter Verwendung einer PC-1 Tabelle unterzogen. Danach werden Daten des Verschlüsselungscode auswählbarer Permutation PC-2 (**908**) unter Verwendung einer PC-2 Tabelle unterzogen. Bei der nächsten Stufe wird jeder Satz von 28 Bits des Verschlüsselungscode, der gemäß einer RS Tabelle gerundet wird, verwendet.

[0124] In dieser Ausführungsform werden vor dem IP Prozess der Umwandlungsprozess a (**901**) zum Umwandeln eines Klartextes, Umwandlungsprozess b (**904**) zum Umwandeln eines Verschlüsselungscode, und letztendlich Rückwandlungsprozess (**916**) zusätzlich ausgeführt. Der Umwandlungsprozess a (**901**) wandelt einen Klartext so um, dass der umgewandelte Klartext später verarbeitet wird und so, dass der Klartext selbst nicht durch den IP Prozess (**902**) und f Funktionsprozess (**903**) verarbeitet wird. Deswegen wird es schwierig, Daten des Klartextes aus der Stromschwingungsform während des Datenprozesses abzuleiten. Der Umwandlungsprozess b (**904**) wandelt einen Verschlüsselungscode so um, dass der umgewandelte Verschlüsselungscode später verarbeitet wird und so, dass der Verschlüsselungscode selbst nicht durch den PC-1 Prozess (**905**), LS Prozess (**907**), PC-2 Prozess (**908**) und f Funktion (**903**) verarbeitet wird. Deswegen wird es schwierig, Daten des Verschlüsselungscode aus der Stromschwingungsform während des Datenprozesses abzuleiten.

[0125] Das Verarbeiten durch die f Funktion ist in **Fig. 10** dargestellt. Dateneingabe zu der f Funktion wird auswählbarer (erweiterter) Permutation unter Verwendung einer auswählbaren E Permutationsmatrix (**1002**) unterzogen. Als Nächstes wird ein logisches Exklusiv-ODER zwischen dem Verschlüsselungscode und einem Ergebnis der auswählbaren Permutation für die Eingabedaten (**1003**) berechnet, ein S Box-Verfahren (**1004**) wird ausgeführt, und ein P Permutationsprozess (**1005**) wird ausgeführt. In dem S Box-Verfahren wird jeder 6-Bit Satz aus 48 Bits extrahiert, welche ein Ergebnis des logischen Exklusiv-ODER bei **1003** sind, um die Reihen- und Spaltenzahlen einer S Box-Tabelle zu erzielen und um 4-Bit Daten zu erzeugen. Die Inhalte der S Box-Tabelle ändern sich mit der Position jedes 6-Bit Satzes. In dem P Permutationsprozess werden die Bit-Positionen von 32 Bits unter Verwendung einer P Permutationstabelle ausgetauscht.

[0126] Der Umwandlungsprozess a (**901**) und Um-

wandlungsprozess b (**902**) sind grundsätzlich dieselben. Mit Bezugnahme auf **Fig. 11** wird der Umwandlungsprozess a zur Umwandlung von Daten eines Klartextes beschrieben. Stördaten X1 werden zufällig erzeugt. Die Stördaten werden unter Verwendung von einem Zufallszahlengenerator oder einer pseudozufälligen Zahl jedes Mal erzeugt, wenn ein DES Verschlüsselungsprozess (oder Entschlüsselungsprozess) durchgeführt (**1102**) wird. Für jeden Prozess werden daher unterschiedliche Stördaten verwendet. Als Nächstes wird ein XOR (logisches Exklusiv-ODER) zwischen den Stördaten X1 und einem Klartext P1 zur Erzeugung eines umgewandelten Klartextes PX1 (**1103**) berechnet. Obwohl ein Klartext von DES 64 Bits aufweist, kann die Zufallszahl entweder 64 Bits oder 8 Bits haben. In diesem Fall, wenn die Bitanzahl der Zufallszahl kleiner als 64 Bits ist, ist es notwendig, sie zu erweitern, um die Stördaten X1 von 64 Bits zu erhalten. Wenn die erzeugte Zufallszahl 8 Bits aufweist, kann diese Zahl achtmal zur Erzeugung der Stördaten X1 von 64 Bits wiederholt werden. Weil das logische Exklusiv-ODER (XOR) zur Umwandlung verwendet wurde, erzeugt XOR zwischen den Stördaten X1 und dem umgewandelten Klartext PX1 den Klartext P.

[0127] Der Umwandlungsprozess b (**904**) zum Umwandeln von Daten eines Verschlüsselungscode ist in **Fig. 36** dargestellt. Ein Unterschied zu der in **Fig. 11** dargestellten Ausführungsform ist derjenige, dass ein Verschlüsselungscode K und Stördaten X2 anstelle eines Klartextes und Stördaten X1 verwendet werden. Ein Verschlüsselungscode von DES weist 64 Bits auf, genauso wie der eines Klartextes. Mit dem Umwandlungsprozess b wird ein umgewandelter Verschlüsselungscode KX2 erzeugt.

[0128] Als Nächstes wird der IP Prozess (**902**) beschrieben. In dem IP Prozess werden die Positionen von einem Klartext, der 64 Bits aufweist, unter Verwendung einer Tabelle, die in **Fig. 37** dargestellt ist, ausgetauscht. Gemäß dieser Tabelle wird das erste Bit einer Ausgabe mit dem 58sten Bit einer Eingabe ausgetauscht, das zweite Bit der Ausgabe wird mit dem 50sten Bit der Eingabe ausgetauscht, ..., und das 64ste Bit der Ausgabe wird mit dem 7ten Bit der Eingabe getauscht. Der IP Prozess dieser Ausführungsform wird mit Bezugnahme auf **Fig. 12** beschrieben. Zuerst wird der umgewandelte Klartext PX1 dem IP Prozess zur Erzeugung eines IP verarbeiteten und umgewandelten Klartextes PX1IP (**1202**) unterzogen. Bit-Tausch wird gemäß der Tabelle, die in **Fig. 37** gezeigt ist, durchgeführt. Als Nächstes werden die Stördaten X1 ebenfalls dem IP Prozess zur Erzeugung von IP verarbeiteten und umgewandelten Stördaten X1IP (**1203**) unterzogen. Ein logisches Exklusiv-ODER zwischen dem IP verarbeiteten und umgewandelten Klartext PX1IP und IP verarbeiteten und umgewandelten Stördaten X1IP kann ein Ergebnis von IP verarbeitetem Klartext erzeugen. Dieses ist so, weil eine Beziehung beibehalten wird, in welcher die Stördaten sich in einer dem umgewan-

deltet Klartext PX1 ähnlichen Weise bewegen, aufgrund einer Bit-Bewegung des IP Prozesses, und somit erzeugt das logische Exklusiv-ODER auf Basis einer Ein-Bit-Einheit die korrekten Daten. Die niedrigeren 32 Bits des Ergebnisses von dem IP Prozess werden in einer ersten Stufe der f Funktion (**903**) und einer zweiten Stufe des logischen Exklusiv-ODER verwendet, wohingegen die höheren 32 Bits für das logische Exklusiv-ODER (**909**) eingegeben werden.

[0129] In dem IP Prozess unterscheiden sich die Werte von Bits des umgewandelten Klartextes PX1 von den Werten von Bits des ursprünglichen Klartextes. Deswegen ist es schwierig, die Daten des ursprünglichen Klartextes von der Stromschwingungsform während des IP Prozesses abzuleiten. Je höher die Anzahl von „1“ Bits, desto größer der Stromverbrauch. Jedoch steht die Anzahl von „1“ Bits des umgewandelten Klartextes überhaupt nicht mit der Anzahl der „1“ Bits des ursprünglichen Klartextes in Zusammenhang, und somit ist es schwierig, die Daten des ursprünglichen Klartextes abzuleiten. Wie vorstehend: weil der Klartext unter Verwendung von Stördaten umgewandelt wird, ist es schwierig, die ursprünglichen Daten abzuleiten, selbst wenn die Stromschwingungsform während des Verarbeitens überwacht wird.

[0130] Mit dem PC-1 Prozess wird eine PC-1 Umwandlungstabelle, die in **Fig. 38** gezeigt ist, verwendet, ein Verschlüsselungscode von 64 Bits wird zu einem Verschlüsselungscode von 56 Bits durch Entfernen der Paritätsbits von 8 Bits verändert, und die Reihenfolge von 56 Bits wird getauscht. Die in **Fig. 38** gezeigte Tabelle wird ähnlich wie die in **Fig. 37** gezeigte Tabelle verwendet. Ein logisches Exklusiv-ODER zwischen dem PC-1 verarbeiteten und umgewandelten Verschlüsselungscode KX2PC1 und PC-1 verarbeiteten und umgewandelten Stördaten X2PC1 kann einen korrekten, PC-1 verarbeiteten und umgewandelten Verschlüsselungscode erzeugen.

[0131] Mit dem LS Prozess wird der Verschlüsselungscode von 56 Bits, der durch den PC-1 Prozess erzeugt wurde, in rechte 28 Bits und linke 28 Bits geteilt, welche dann um ein Bit oder um zwei Bits gemäß der LS Tabelle nach links verschoben werden. Diese Ausführungsform wird mit Bezugnahme auf **Fig. 15** beschrieben. Zuerst wird bei **1502** der PC-1 verarbeitete und umgewandelte Verschlüsselungscode KX2PC1 dem LS Prozess zur Erzeugung eines PC-1 und LS verarbeiteten und umgewandelten Verschlüsselungscodes KX2PC1LS unterzogen. Bei **1503** werden die PC-1 verarbeiteten Stördaten X2PC1 dem LS Prozess zur Erzeugung von PC-1 und LS verarbeiteten Stördaten X2PC1LS unterzogen. Weil in dem LS Prozess ebenfalls Bit-Positionstausch verwendet wird, kann ein logisches Exklusiv-ODER zwischen dem PC-1 und LS verarbeiteten und umgewandelten Verschlüsselungscode KX2PC1LS und PC-1 und LS verarbeiteten Stördaten X2PC1LS einen korrekten, LS verarbeiteten Ver-

schlüsselungscode erzeugen. Weil die Stördaten ebenfalls in dem LS Prozess verwendet werden und die Daten des Verschlüsselungscodes, die tatsächlich verarbeitet werden, sich von dem ursprünglichen Verschlüsselungscode unterscheiden, ist es schwierig, den Verschlüsselungscode abzuleiten, selbst wenn die Stromschwingungsform überwacht wird.

[0132] In dem PC-2 Prozess wird eine Reduzierungsspermutation zum Verändern der 56-Bit Daten, die durch den LS Prozess erzeugt wurden, auf 48-Bit Daten gemäß einer PC-2 Tabelle ausgeführt. Bei **1402** wird der PC-1 und LS verarbeitete und umgewandelte Verschlüsselungscode KX2PC1LS dem PC-2 Prozess zur Erzeugung eines PC-1, LS und PC-2 verarbeiteten und umgewandelten Verschlüsselungscodes KX2PC1LSPC2 unterzogen. Bei **1403** werden die PC-1 und LS verarbeiteten Stördaten X2PC1LS dem PC-2 Prozess zur Erzeugung von PC-1, LS und PC-2 verarbeiteten Stördaten X2PC1LSPC2 unterzogen. Grundsätzlich wird in diesem PC-2 Prozess die Tabelle für Permutation verwendet, so dass er grundsätzlich derselbe ist wie der PC-1 Prozess.

[0133] Als Nächstes wird der Prozess für die f Funktion **903** beschrieben. Wie in **Fig. 10** dargestellt umfasst die f Funktion einen auswählbaren Permutation E Prozess (**1002**), ein logisches Exklusiv-ODER (**1003**) zwischen einem Verschlüsselungscode und einem Ausführungsergebnis von auswählbarer Permutation, ein S Box-Verfahren (**1004**), und einen P Permutationsprozess (**1005**).

[0134] Der auswählbare Permutation E Prozess wird mit Bezugnahme auf **Fig. 16** beschrieben. Ähnlich wie der IP Prozess wird in dem auswählbaren Permutation E Prozess die Reihenfolge von Bits unter Verwendung einer Permutationstabelle, die in **Fig. 28** gezeigt ist, getauscht. Bei **1602** wird der IP verarbeitete und umgewandelte Klartext PXIP dem auswählbaren Permutation E Prozess zur Erzeugung eines IP verarbeiteten, E permutierten und umgewandelten Klartextes PXIPE unterzogen. Bei **1603** werden die IP verarbeiteten Stördaten XIP dem auswählbaren Permutation E Prozess zur Erzeugung von IP verarbeiteten, E permutierten Stördaten XIPE unterzogen. Ähnlich dem IP Prozess und dem PC-1 Prozess kann ein logisches Exklusiv-ODER zwischen dem IP verarbeiteten, E permutierten und umgewandelten Klartext PXIPE und IP verarbeiteten, E permutierten Stördaten XIPE einen korrekten, IP verarbeiteten, E permutierten Klartext erzeugen. Weil sich die Werte von Bits, die unter Verwendung der Permutation E Tabelle getauscht wurden, von jenen der ursprünglichen Bits unterscheiden, ist es schwierig, die korrekten Daten abzuleiten, selbst wenn die Stromschwingungsform während dieses Prozesses überwacht wird.

[0135] Als Nächstes wird der zweite Prozess (**1003**) der f Funktion, d. h. ein logisches Exklusiv-ODER zwischen dem Verschlüsselungscode und einem Ausführungsergebnis von auswählbarer Permutation, mit Bezugnahme auf **Fig. 17**, beschrieben. Bei

1702 wird ein XOR zwischen dem IP verarbeiteten, E permutierten und umgewandelten Klartext PXIPE, erzeugt aus dem Klartext, und dem PC-1, LS und PC-2 verarbeiteten und umgewandelten Verschlüsselungscode KX2PC1LSPC2, erzeugt von dem Verschlüsselungscode, zur Erzeugung von 48-Bit S Box Eingabedaten SinputX, welche als eine Eingabe für das S Box-Verfahren verwendet werden, berechnet. Als Nächstes wird bei **1703** ein XOR zwischen den IP verarbeiteten, E permutierten Stördaten XIPE, erzeugt von den Stördaten für den Klartext, und den PC-1, LS und PC-2 verarbeiteten Stördaten X2PC1LSPC2, erzeugt von den Stördaten für den Verschlüsselungscode, zur Erzeugung von S Box Eingabedaten-Stördaten XSinput, welche als die Stördaten für die S Box Eingabedaten SinputX verwendet werden, berechnet. Beruhend auf den Charakteristika eines logischen Exklusiv-ODER, können die S Box Eingabedaten-Stördaten XSinput durch ein XOR zwischen den zwei Sätzen von Stördaten (den IP verarbeiteten, E permutierten Stördaten XIPE und den PC-1, LS und PC-2 verarbeiteten Stördaten X2PC1LSPC2) erzeugt werden. Dieses wird unter Verwendung eines einfachen Beispiels erklärt. Ein Klartext ist durch P dargestellt, ein Verschlüsselungscode ist durch K dargestellt, ein umgewandelter Klartext ist durch PX1 dargestellt, und ein umgewandelter Verschlüsselungscode ist durch KX2 dargestellt. Die Beziehungen zwischen ihnen werden durch die folgenden Gleichungen (63) und (64) gegeben, wobei X1 und X2 Stördaten für jeweils den Klartext und Verschlüsselungscode sind:

$$PX1 = P \text{ xor } X1$$

(63)

$$KX2 = K \text{ xor } X2$$

(64)

[0136] Wenn ein Ausführungsergebnis des logischen Exklusiv-ODER zwischen P und K durch Z dargestellt ist, dann ist die Beziehung zwischen Z und einem Ausführungsergebnis Z1 des logischen Exklusiv-ODER zwischen PX1 und PX2 gegeben durch:

$$Z = P \text{ xor } K$$

(65)

$$Z1 = PX1 \text{ xor } KX2 = (P \text{ xor } X1) \text{ xor } (K \text{ xor } X2) = P \text{ xor } X1 \text{ xor } K \text{ xor } X2 = (P \text{ xor } K) \text{ xor } (X1 \text{ xor } X2) = Z \text{ xor } (X1 \text{ xor } X2)$$

(66)

[0137] Durch die obige Gleichung kann vorausgesetzt werden, dass als die Stördaten zur Wiederherstellung der korrekten Daten von Z1 das logische Exklusiv-ODER zwischen P und K Stördaten verwendet werden kann. In dem logischen Exklusiv-ODER Prozess zwischen dem Verschlüsselungscode und dem Ausführungsergebnis von auswählbarer Permutation können die S Box Eingabedaten-Stördaten XSinput als die Stördaten für die S Box Eingabedaten SinputX

verwendet werden, wobei die S Box Eingabedaten-Stördaten XSinput durch ein XOR zwischen den IP verarbeiteten, E permutierten Stördaten XIPE, erzeugt von den Stördaten für den Klartext, und den PC-1, LS und PC-2 verarbeiteten Stördaten X2PC1LSPC2, erzeugt von den Stördaten für den Verschlüsselungscode, erzeugt werden.

[0138] Als Nächstes wird das S Box-Verfahren mit Bezugnahme auf **Fig. 18** beschrieben. Acht S Boxen werden ausgebildet, die jeweils aus 6 Bits der S Box Eingabedaten SinputX bestehen. Die erste S Box, die in DES verwendet wird, ist in **Fig. 25** gezeigt. Obwohl das Format der acht S Boxen jeweils dasselbe ist, sind die Daten in jedem Feld unterschiedlich. In dem S Box-Verfahren für jede S Box werden zuerst die Sub-Daten SubSinputX(i) von 6 Bits von den S Box Eingabedaten Sinput an der oberen i-ten (**1805**) hergeleitet. Ein XOR wird zwischen diesen SubSinputX(i) und Adressen-Stördaten Xsa(i) für eine S Box-Tabelle, die vorher von der S Box umgewandelt wurden, zur Erzeugung von SubSinputXXsa(i) (**1806**) berechnet. Ein XOR wird zwischen diesen SubSinputXXsa(i) und sechs Bits von den S Box Eingabedaten-Stördaten XSinput an der oberen i-ten zur Erzeugung von SubSinputXsa(i) (**1807**) berechnet. Diese SubSinputXsa(i) sind die XOR berechneten Daten der Adressen-Stördaten Xsa(i) und der korrekten Adressendaten, die zur Herleitung der i-ten S Box verwendet werden sollen. Weil das logische Exklusiv-ODER (XOR) zwischen SubSinputX(i) und XSinput(i) korrekte Daten wiederherstellt, wird ein XOR zwischen SubSinputX(i) und Xsa(i) berechnet, und das XOR zwischen dem Ergebnis und XSinput(i) wird berechnet. Mit diesem Verfahren ist es nicht notwendig, die ursprünglichen Daten zu verarbeiten, so dass es schwierig ist, dass die ursprünglichen Daten von der Stromschwingungsform abgeleitet werden. Als Nächstes wird eine Adresse von der umgewandelten S Box-Tabelle unter Verwendung von SubSinputXsa(i) (**1808**) berechnet. Weil die Adresse zum Zugreifen auf die ursprüngliche S Box-Tabelle umgewandelt wurde, ist es ebenfalls notwendig, die Tabelle umzuwandeln. Unter Verwendung der berechneten Adresse werden S Box Ausgabedaten SoutputX3(i) von der umgewandelten S Box-Tabelle S(i) (**1809**) wiedergewonnen. In demselben Moment werden Stördaten X3(i) für die S Box Ausgabedaten SoutputX3(i) wiedergewonnen (**1810**). Nachdem die acht S Boxen verarbeitet worden sind, werden Daten von SoutputX3(i) und X3(i) für i = 1 bis 8 zur Erzeugung von SoutputX3 und X3 miteinander verbunden. SoutputX3 werden als die Prozessdaten verwendet und X1 werden als die Stördaten in den nachfolgenden Prozessen verwendet.

[0139] Als Nächstes wird ein Verfahren zur Erzeugung der umgewandelten S Box-Tabelle mit Bezugnahme auf **Fig. 23** und **24** beschrieben. Adressen-Stördaten Xsa(i) und Daten-Stördaten X3(i) werden von einer S(i) Box (**2306**) erzeugt. Xsa(i) weist 6 Bits und X3(i) weist 4 Bits auf. Die Stördaten X3 wei-

sen 32 Bits auf, gesammelt von acht X3(i), die jeweils 4 Bits aufweisen. Als Nächstes wird eine Routine zur Ausbildung einer umgewandelten S(i) Box-Tabelle aufgerufen (**2307**). Die Routine zur Ausbildung der i-ten umgewandelten S Box-Tabelle wird mit Bezugnahme auf **Fig. 24** beschrieben. Zur Bezeichnung einer Reihenzahl wird k verwendet und zur Bezeichnung einer Spaltenzahl wird l verwendet. Der Prozess für die k-te Reihe und l-te Spalte ist von **2408** bis **2413** dargestellt. Die erste S Box-Tabelle ist in **Fig. 25** gezeigt. Zuerst werden Daten d bei der k-ten Reihe und l-ten Spalte von der i-ten ursprünglichen S Box (**2408**) aufgenommen. Ein logisches Exklusiv-ODER zwischen den Daten d und den Stördaten X3(i) wird zur Erzeugung von Daten d2 (**2409**) berechnet. Wenn die Stördaten „7“ sind, werden die umgewandelten Daten der ursprünglichen S Box Daten von 2504 in **2604** angezeigt. Diese Operation wird für alle Felder zur Erlangung einer umgewandelten S Box-Tabelle, gezeigt in **Fig. 26**, ausgeführt. Diese in **Fig. 26** gezeigte Tabelle wird durch logische Exklusiv-ODER Operationen zwischen den ersten S Box-Daten und den Stördaten „7“ ausgebildet.

[0140] Als Nächstes wird die Adresse gestört. Zuerst wird Xsa1 aus zwei Bits ausgebildet, die das obere erste Bit und das untere erste Bit von Xsa(i) umfassen, und Xsa2 wird aus vier Bits ausgebildet, die die Bits von dem oberen zweiten Bit bis zu dem oberen fünften Bit von Xsa(i) umfassen. Dieser Prozess wird von dem S Box Adressenberechnungsverfahren verursacht. Durch Darstellen der Reihen- und Spaltenzahlen der Tabelle, die in **Fig. 26** gezeigt ist, durch k und l, werden logische Exklusiv-ODER Operationen zwischen der Xsa1 und Xsa2 für entsprechende Reihe und Spalte ausgeführt (**2412**). Durch Darstellen der neuen Reihen- und Spaltenzahlen durch k2 und l2, werden die Daten d2 in der i-ten umgewandelten S Box-Tabelle S(i) in der k2-ten Reihe und l2-ten Spalte (**2413**) gespeichert. Ein Beispiel dieses Prozesses ist in **Fig. 27** dargestellt. Die in **Fig. 27** gezeigte Tabelle wird unter Verwendung von Stördaten „2“ und „9“ für die Reihe und Spalte, die in **Fig. 26** gezeigt ist, ausgebildet. In dieser Tabelle wird, aus Gründen der Vereinfachung, die Position von Daten in jeder Reihe und jeder Spalte nicht verändert, nur die Reihen- und Spaltenzahlen werden verändert. Die bei 2504 angezeigten Daten „12“ in der dritten Reihe und ersten Spalte der in **Fig. 25** gezeigten Tabelle werden zu der ersten Reihe und achten Spalte, in **Fig. 27**, bewegt und ihr Wert wird auf „11“ geändert. In diesem Beispiel sind die Daten-Stördaten „7“, und die Adressen-Stördaten haben die Reihenzahl von „2“ und die Spaltenzahl von „9“. Auf diese Weise werden die acht S Boxen umgewandelt. In dieser Ausführungsform wird dieser Prozess in der ersten Stufe von DES ausgeführt. Die umgewandelten S Boxen werden in sechzehn Stufen von DES verwendet.

[0141] Nach dem S Box-Verfahren werden 32-Bit SoutputX3 als Prozessdaten verwendet, und 32-Bit

X3 werden als Stördaten verwendet, welche zu dem Permutation P Prozess (**1005**) geliefert werden, welcher der letzte Prozess der f Funktion ist. Der Permutation P Prozess wird mit Bezugnahme auf **Fig. 19** beschrieben. SinputX3, geliefert von dem S Box-Verfahren, wird dem Permutation P Prozess zur Erzeugung von SinputX3P (**1902**) unterzogen. Die Stördaten X3 für SinputX3 werden dem Permutation P Prozess zur Erzeugung von X3P (**1903**) unterzogen. Eine Tabelle, die für den Permutation P Prozess verwendet wird, ist in **Fig. 29** gezeigt. Diese Tabelle wird ähnlich wie die IP Prozess Tabelle verwendet.

[0142] Nachdem der f Funktionsprozess abgeschlossen ist, wird ein XOR zwischen einem Ergebnis von dem Permutation P Prozess und einem Ergebnis in der vorhergehenden Stufe berechnet (**909**, **914**). Insbesondere wird ein XOR zwischen SinputX3P, erhalten durch den Permutation P Prozess für das S Box Verarbeitungsergebnis, und einem Ergebnis in der vorhergehenden Stufe (**2002**) berechnet. Ein XOR wird zwischen X3P und den Stördaten X in der vorhergehenden Stufe (**2003**) berechnet. Dieser XOR Prozess ist derselbe wie derjenige (**1701**) zwischen dem Ergebnis des auswählbaren Permutation E Prozesses und dem Verschlüsselungscode.

[0143] Bei DES wird der IP⁻¹ Prozess (**915**) in der letzten Stufe ausgeführt. Dieser Prozess ist in **Fig. 21** dargestellt. Der IP-1 Prozess ist ein Bit-Positionentauschprozess, ähnlich dem IP Prozess, unter Verwendung einer IP-1 Tabelle anstelle der IP Tabelle (**2102**). Das Verarbeitungsergebnis, das durch die oben beschriebenen Prozesse erhalten wird, wird dem IP-1 Prozess unterzogen, und die Stördaten X werden ebenfalls dem IP-1 Prozess (**2103**) unterzogen.

[0144] Abschließend, zur Wiederherstellung des korrekten Verarbeitungsergebnisses, wird der Rückwandlungsprozess (**916**) ausgeführt, welcher in **Fig. 22** dargestellt ist. Ein XOR zwischen dem IP-1 Verarbeitungsergebnis und den IP-1 verarbeiteten Stördaten X erzeugt ein korrektes Ergebnis. Das nicht umgewandelte korrekte Verarbeitungsergebnis kann in dieser Stufe das erste Mal erhalten werden.

[0145] Das Verschleiern der Prozessdaten ist vorstehend beschrieben worden. Es gibt den Fall, dass die Stördaten ebenfalls verschleiert werden müssen. Das grundsätzliche Konzept ist, die Stördaten durch ein logisches Exklusiv-ODER zwischen den Stördaten und Stördaten XR für Störung umzuwandeln. In diesem Fall ist XR feststehend und XRo für die Rückwandlung wird im Vorhinein durch Berechnung des Bit-Positionstausches oder dergleichen erhalten. Wenn die Stördaten notwendig werden, werden die ursprünglichen Stördaten unter Verwendung von XRo erhalten. Zuerst wird dieser Prozess beschrieben, indem Stördaten für den Verschlüsselungscode als ein Beispiel genommen werden. Der Prozess, der in **Fig. 30** dargestellt ist, ist ein Stördaten-Umwandlungsprozess durch ein logisches Exklusiv-ODER

zwischen den Stördaten für einen Verschlüsselungscode und den Stördaten XR für Störung. Nachdem der Umwandlungsprozess b (**3601**) die Stördaten X2 erzeugt, wird der Stördaten-Umwandlungsprozess, der in **Fig. 30** gezeigt ist, ausgeführt. Die Stördaten X2 für den Verschlüsselungscode werden dem PC-1 Prozess, LS Prozess und PC-2 Prozess unterzogen. Diese Prozesse führen einen Bit-Tausch an vorgegebenen Bit-Positionen durch. Deswegen werden, für den vorgegebenen Wert XR, die Stördaten XRo, die den Prozessen bis zu dem PC-2 Prozess unterzogen wurden, berechnet und im Vorhinein gespeichert (**3102** bis **3105**). Nach dem PC-2 Prozess, wird ein XOR (**3202**) zwischen den PC-1, LS, PC-2 verarbeiteten Stördaten X2PC1LSPC2, erzeugt bei **1403**, berechnet, und die gespeicherten Stördaten XRo können korrekte, PC-1, LS, PC-2 verarbeitete Stördaten X2PC1LSPC2 erzeugen. Die Stördaten können mit diesen Prozessen ebenfalls verschleiert werden. Dieselben Daten können als die Stördaten XR für Störung verwendet werden, und als die verarbeiteten, umgewandelten Stördaten XRo.

[0146] Die Ausführungsformen für DES betreffen Verschlüsselung. Weil fast derselbe DES Algorithmus ebenfalls für Entschlüsselung verwendet wird, können die Ausführungsformen ebenfalls auf Entschlüsselung angewendet werden, wobei die Ausführungsformen kaum verändert werden. In anderen Verschlüsselungsalgorithmen als DES werden viele Permutationsprozesse, Substitutionsprozesse und Modularrechnungen verwendet. Deswegen kann die Erfindung auf solche Algorithmen angewendet werden, um es schwierig zu gestalten, Daten zu verändern und ursprüngliche Daten durch Beobachtung der Stromschwingungsformen abzuleiten.

[0147] Gemäß der vorliegenden Erfindung können Daten, die durch einen IC-Kartenchip verarbeitet werden sollen, so umgewandelt werden, dass es schwierig ist, Prozesse und einen Verschlüsselungscode durch Beobachtung von Stromschwingungsformen abzuleiten.

Patentansprüche

1. Informationsverarbeitungseinrichtung mit einem Speicher mit einer Programmspeichereinheit und einer Datenspeichereinheit, einer Zentralverarbeitungseinheit zum Ausführen eines Datenprozesses entsprechend dem Programm, wobei das Programm ein oder mehrere Datenverarbeitungseinrichtungen aufweist, deren jede ein Prozessbefehl ist, um der Zentralverarbeitungseinheit einen Ausführungsbefehl zu erteilen, und einer Eingabedaten-Verarbeitungseinrichtung, wobei eine Datenverarbeitungseinrichtung Eingabedaten verarbeitet und die verarbeiteten Daten ausgibt, gekennzeichnet durch eine Datenumwandlungseinrichtung (**406**) zum Umwandeln von Eingabedaten D1 unter Verwendung von Stördaten Xi zur Erzeugung umgewandelter Da-

ten H1, eine umgewandelte Daten verarbeitende Einrichtung (**408**) zum Ausführen eines Operationsprozesses OP1 an den umgewandelten Daten H1 anstelle des von der Eingabedaten-Verarbeitungseinrichtung auszuführenden Operationsprozesses OP1 für die Eingabedaten D1, um verarbeitete und umgewandelte Daten H2 zu erzeugen, eine Stördaten-Verarbeitungseinrichtung (**403**) zum Ausführen des Operationsprozesses OP1 an den Stördaten Xi zur Erzeugung verarbeiteter Stördaten Xo und eine Datenrückwandlungseinrichtung (**410**) zum Ausführen eines Operationsprozesses OP2 an den verarbeiteten und umgewandelten Daten H2 unter Verwendung der verarbeiteten Stördaten Xo zur Erzeugung verarbeiteter Daten D2 als Ergebnis des Operationsprozesses OP1 an den Eingabedaten D1.

2. Informationsverarbeitungseinrichtung mit einem Speicher mit einer Programmspeichereinheit und einer Datenspeichereinheit und einer Zentralverarbeitungseinheit zum Ausführen eines Datenprozesses entsprechend dem Programm, wobei das Programm ein oder mehrere Datenverarbeitungseinrichtungen enthält, gekennzeichnet durch eine Einrichtung (**506**) zum Ausführen eines vorgegebenen Operationsprozesses OP1 an ersten Eingabedaten D1 unter Verwendung erster Stördaten X1i zur Erzeugung umgewandelter Daten H1 der ersten Eingabedaten D1, ohne Ausführung des Operationsprozesses OP1 an den ersten Eingabedaten D1 zur Erzeugung erster verarbeiteter Daten D2, eine Einrichtung (**508**) zum Ausführen des Operationsprozesses OP1 oder eines von diesem verschiedenen weiteren Operationsprozesses OP1' an den umgewandelten Daten H1 zur Erzeugung verarbeiteter und umgewandelter Daten H2, eine Einrichtung (**503**) zum Ausführen entweder des Operationsprozesses OP1 oder des weiteren Operationsprozesses OP1' an den ersten Stördaten X1i zur Erzeugung erster verarbeiteter Stördaten X1o, eine Einrichtung (**510**) zum Erzeugen verarbeiteter und umgewandelter Daten H3 durch Umwandeln der verarbeiteten und umgewandelten Daten H2 unter Verwendung zweiter Stördaten X2i, eine Einrichtung (**512**) zum Ausführen entweder des Operationsprozesses OP1 oder des weiteren Operationsprozesses OP1' an den verarbeiteten und umgewandelten Daten H3 zur Erzeugung verarbeiteter und umgewandelter Daten H4, eine Einrichtung (**518**) zum Ausführen entweder des Operationsprozesses OP1 oder des weiteren Operationsprozesses OP1' an den zweiten Stördaten X2i zur Erzeugung zweiter verarbeiteter Stördaten X2o, eine Einrichtung (**520**) zum Verarbeiten der verarbeiteten und umgewandelten Daten H4 unter Verwendung der zweiten verarbeiteten Stördaten X2o zur Erzeugung verarbeiteter und umgewandelter Daten H5,

und
eine Einrichtung **(514)** zum Verarbeiten der verarbeiteten und umgewandelten Daten H5 unter Verwendung der ersten verarbeiteten Stördaten X1o zur Erzeugung der ersten verarbeiteten Daten D2.

3. Informationsverarbeitungseinrichtung mit einem Speicher mit einer Programmspeichereinheit und einer Datenspeichereinheit und einer Zentralverarbeitungseinheit zum Ausführen eines Datenprozesses entsprechend dem Programm, wobei das Programm ein oder mehrere Datenverarbeitungseinrichtungen enthält, gekennzeichnet durch
eine Einrichtung **(607)** zum Ausführen eines vorgegebenen Operationsprozesses OP1 an ersten Eingabedaten D1 unter Verwendung erster Stördaten X1i zur Erzeugung umgewandelter Daten H1 der ersten Eingabedaten D1, ohne Ausführung des Operationsprozesses OP1 an den ersten Eingabedaten D1 zur Erzeugung erster verarbeiteter Daten D3,
eine Einrichtung **(610)** zum Ausführen des Operationsprozesses OP1 oder eines von diesem verschiedenen Operationsprozesses OP1' an den umgewandelten Daten H1 zur Erzeugung verarbeiteter und umgewandelter Daten H2,
eine Einrichtung **(603)** zum Ausführen entweder des Operationsprozesses OP1 oder des weiteren Operationsprozesses OP1' an den ersten Stördaten X1i zur Erzeugung erster verarbeiteter Stördaten X1o,
eine Einrichtung **(610)** zum Erzeugen verarbeiteter und umgewandelter Daten H3 durch Umwandeln der verarbeiteten und umgewandelten Daten H2 unter Verwendung zweiter Stördaten X2i,
eine Einrichtung **(605)** zum Ausführen eines Rückwandlungsprozesses an den verarbeiteten und umgewandelten Daten H3 unter Verwendung der ersten verarbeiteten Stördaten X1o zur Erzeugung verarbeiteter und umgewandelter Daten H4,
eine Einrichtung **(614)** zum Ausführen entweder eines vorgegebenen Operationsprozesses OP2 oder eines von diesem verschiedenen weiteren Operationsprozesses OP2' an den verarbeiteten und umgewandelten Daten H4 zur Erzeugung verarbeiteter und umgewandelter Daten H5, ohne Ausführung des Operationsprozesses OP2 an den ersten verarbeiteten Daten D3 zur Erzeugung der ersten und der zweiten verarbeiteten Daten D2,
eine Einrichtung **(620)** zum Ausführen entweder des Operationsprozesses OP2 oder des weiteren Operationsprozesses OP2' an den zweiten Stördaten X2i zur Erzeugung zweiter verarbeiteter Stördaten X2o, und
eine Einrichtung **(616)** zum Verarbeiten der verarbeiteten und umgewandelten Daten H2 unter Verwendung der zweiten verarbeiteten Stördaten X2o zur Erzeugung der verarbeiteten Daten D2.

4. Informationsverarbeitungseinrichtung, gekennzeichnet durch

eine Einrichtung **(812)** zum Ausführen eines vorgegebenen Operationsprozesses OP1 an ersten Eingabedaten D1 unter Verwendung erster Stördaten X1i zur Erzeugung umgewandelter Daten H1 der ersten Eingabedaten D1, ohne Ausführung des Operationsprozesses OP1 an den ersten Eingabedaten D1 zur Erzeugung erster verarbeiteter Daten D3,
eine Einrichtung **(814)** zum Ausführen des Operationsprozesses OP1 oder eines von diesem verschiedenen weiteren Operationsprozesses OP1' an den umgewandelten Daten H1 zur Erzeugung verarbeiteter und umgewandelter Daten H2,
eine Einrichtung **(803)** zum Ausführen entweder des Operationsprozesses OP1 oder des weiteren Operationsprozesses OP1' an den ersten Stördaten X1i zur Erzeugung erster verarbeiteter Stördaten X1o,
eine Einrichtung **(816)** zum Erzeugen verarbeiteter und umgewandelter Daten H3 durch Umwandeln der verarbeiteten und umgewandelten Daten H2 unter Verwendung zweiter Stördaten X2i,
eine Einrichtung **(818)** zum Ausführen eines Rückwandlungsprozesses an den verarbeiteten und umgewandelten Daten H3 unter Verwendung der ersten verarbeiteten Stördaten X1o zur Erzeugung verarbeiteter und umgewandelter Daten H4,
eine Einrichtung zum Ausführen entweder eines vorgegebenen Operationsprozesses OP2 oder eines von diesem verschiedenen weiteren Operationsprozesses OP2' an den verarbeiteten und umgewandelten Daten H4 zur Erzeugung verarbeiteter und umgewandelter Daten H5, ohne Ausführung des Operationsprozesses OP2 an den ersten verarbeiteten Daten D3 zur Erzeugung der ersten und der zweiten verarbeiteten Daten D2,
eine Einrichtung **(807)** zum Ausführen entweder des Operationsprozesses OP2 oder des weiteren Operationsprozesses OP2' an den zweiten Stördaten X2i zur Erzeugung zweiter verarbeiteter Stördaten X2o,
eine Einrichtung **(809)** zum Erzeugen einheitlicher Stördaten Xo durch Vereinheitlichen der ersten und der zweiten verarbeiteten Stördaten X1o und X2o, und
eine Einrichtung **(820)** zum Ausführen eines Rückwandlungsprozesses an den verarbeiteten und umgewandelten Daten H4 unter Verwendung der einheitlichen Stördaten Xo zur Erzeugung der verarbeiteten Daten D2.

5. Einrichtung nach Anspruch 1 mit ferner
einer Einrichtung **(704)** zum Speichern der verarbeiteten Stördaten Xo, und
einer Einrichtung **(713)** zum Verarbeiten der verarbeiteten und umgewandelten Daten unter Verwendung der gespeicherten verarbeiteten Stördaten Xo zur Erzeugung neuer verarbeiteter umgewandelter Daten.

6. Einrichtung nach einem der vorhergehenden Ansprüche, wobei die Stördaten unter Verwendung einer Zufallszahl erzeugt werden.

7. Einrichtung nach Anspruch 1, wobei der Datenumwandlungsprozess (a) ein logischer Exklusiv-ODER-Prozess oder (b) ein Addierprozess oder (c) ein Subtraktionsprozess oder (d) ein Multiplikationsprozess oder (e) ein Divisionsprozess zwischen den Stördaten X_i und den Eingabedaten D_1 ist und

der Datenrückwandlungsprozess (a) ein logischer Exklusiv-ODER-Prozess oder (b) ein Subtraktionsprozess oder (c) ein Additionsprozess oder (d) ein Divisionsprozess oder (e) ein Multiplikationsprozess zwischen den verarbeiteten Stördaten X_o und den verarbeiteten und umgewandelten Daten H_2 ist.

8. Einrichtung nach Anspruch 1, wobei dann, wenn der Eingabedatenprozess einen Additions/Subtraktionsprozess in einer Modularrechnung enthält, der Datenumwandlungsprozess mit einer Addition/Subtraktion einer Zahl arbeitet, die einen Modulus N mit einer beliebiger. ganzen Zahl multipliziert.

9. Einrichtung nach Anspruch 1, wobei dann, wenn der Eingabedatenprozess einen Multiplikationsprozess in einer Modularrechnung enthält und die Zahlen X und Y die Bedingung $1 = X * Y \bmod N$ erfüllen,

(a) X in der Modularrechnung des Datenumwandlungsprozesses mit einer ganzen Zahl und Y mit einer ganzen Zahl multipliziert wird, sooft die Daten umwandelnde Verarbeitungseinrichtung mit X multipliziert, oder

(b) $(N + 1)/2$ in der Modularrechnung des Datenumwandlungsprozesses mit einer ganzen Zahl multipliziert wird, wobei N als Modulus der Modularrechnung benutzt wird und

2 mit einer ganzen Zahl multipliziert wird, sooft die Daten umwandelnde Verarbeitungseinrichtung in dem Datenrückwandlungsprozess mit $(N + 1)/2$ multipliziert, oder

(c) 2 in der Modularrechnung des Daten umwandelnden Prozesses mit einer ganzen Zahl multipliziert wird und

$(N + 1)/2$ mit einer ganzen Zahl multipliziert wird, sooft die Daten umwandelnde Verarbeitungseinrichtung in dem Datenrückwandlungsprozess mit 2 multipliziert, wobei N als Modulus der Modularrechnung benutzt wird.

10. Einrichtung nach Anspruch 1, wobei der Datenumwandlungsprozess Positionen von Arraydaten regelmäßig ändert und der Datenrückwandlungsprozess auf die durch den Datenumwandlungsprozess geänderten Arraydaten zugreift.

11. Einrichtung nach Anspruch 10, wobei die Positionen der Arraydaten durch ein logisches Exklusiv-ODER zwischen einem Index (Argument) der Arraydaten und einer bestimmten Zahl regelmäßig geändert werden, ein logisches Exklusiv-ODER zwischen dem Index der Arraydaten und der von dem

Datenumwandlungsprozess benutzten Zahl als Index des umgewandelten Datenarrays verwendet wird, und der Datenrückwandlungsprozess entsprechend dem umgewandelten Index auf die Arraydaten zugreift.

12. Einrichtung nach Anspruch 1, wobei der Operationsprozess OP_1 (a) ein datentauschender Permutationsprozess auf der Basis einer 1-Bit-Einheit oder (b) ein datentauschender Substitutionsprozess auf der Basis einer 1-Bit-Einheit oder (c) ein Datentauschprozess unter Verwendung einer Tabelle ist, und wobei der Datenumwandlungsprozess und der Datenrückwandlungsprozess ein logischer Exklusiv-ODER-Prozess an den Daten ist.

13. Informationsverarbeitungseinrichtung mit einer Verarbeitungseinheit zum Verarbeiten von Eingabedaten entsprechend einem Computerprogramm und zur Ausgabe der verarbeiteten Daten, gekennzeichnet durch

eine Einrichtung (**709**) zum Umwandeln von Eingabedaten D_1 unter Verwendung von Stördaten X_i zur Erzeugung umgewandelter Daten H_1 , ohne Ausführung eines vorgegebenen Operationsprozesses OP_1 an den Eingabedaten D_1 zur Erzeugung verarbeiteter Daten D_2 ,

eine Einrichtung (**712**) zum Ausführen des Operationsprozesses OP_1 oder eines von diesem verschiedenen weiteren Operationsprozesses OP_1' an den umgewandelten Daten H_1 zur Erzeugung verarbeiteter und umgewandelter Daten H_2 ,

eine Einrichtung (**701**) zum Ausführen entweder des Operationsprozesses OP_1 oder des weiteren Operationsprozesses OP_1' an den Stördaten X_i zur Erzeugung verarbeiteter Stördaten X_o und

eine Einrichtung (**713**) zum Ausführen eines Datenrückwandlungsprozesses OP_2 an den verarbeiteten und umgewandelten Daten H_2 unter Verwendung der verarbeiteten Stördaten X_o zur Erzeugung der verarbeiteten Daten D_2 als Resultat des Operationsprozesses OP_1 an den Eingabedaten D_1 .

Es folgen 24 Blatt Zeichnungen

FIG. 1

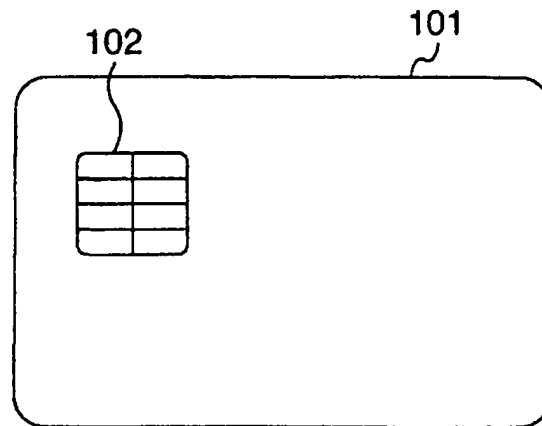


FIG. 2

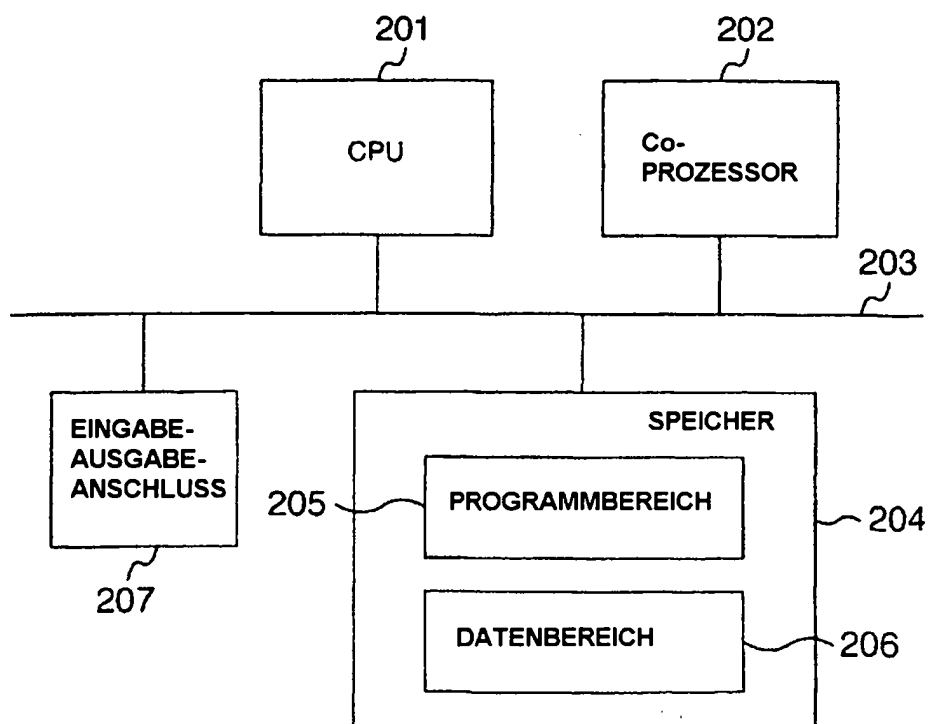


FIG. 3

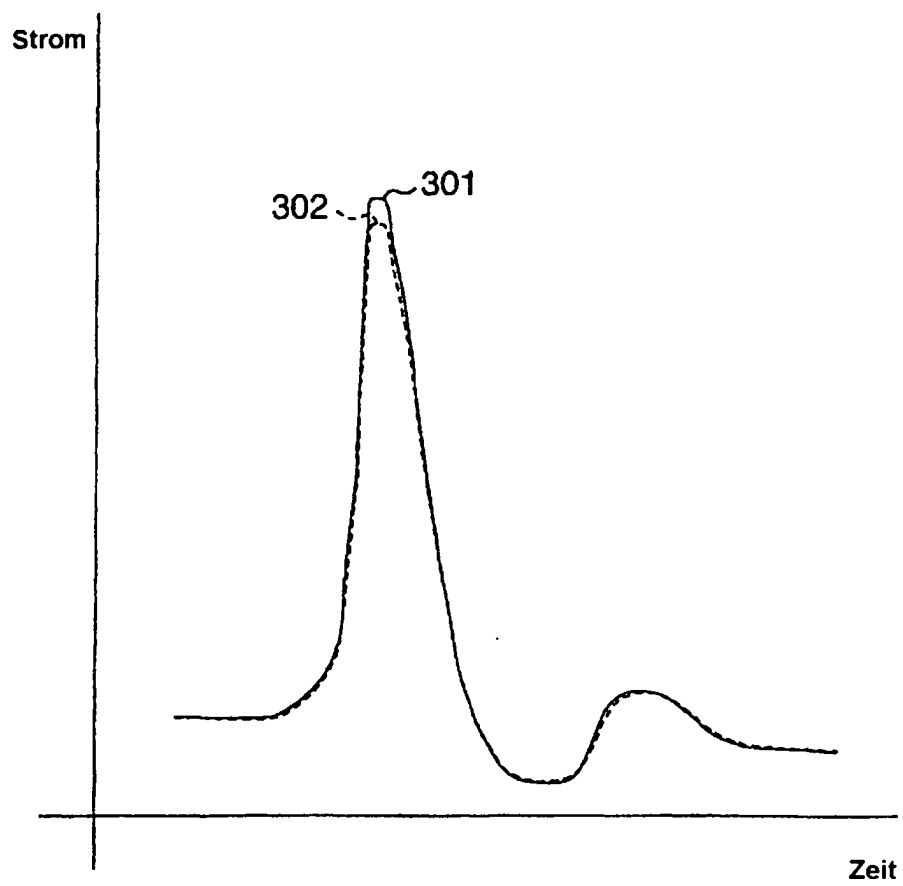


FIG. 4

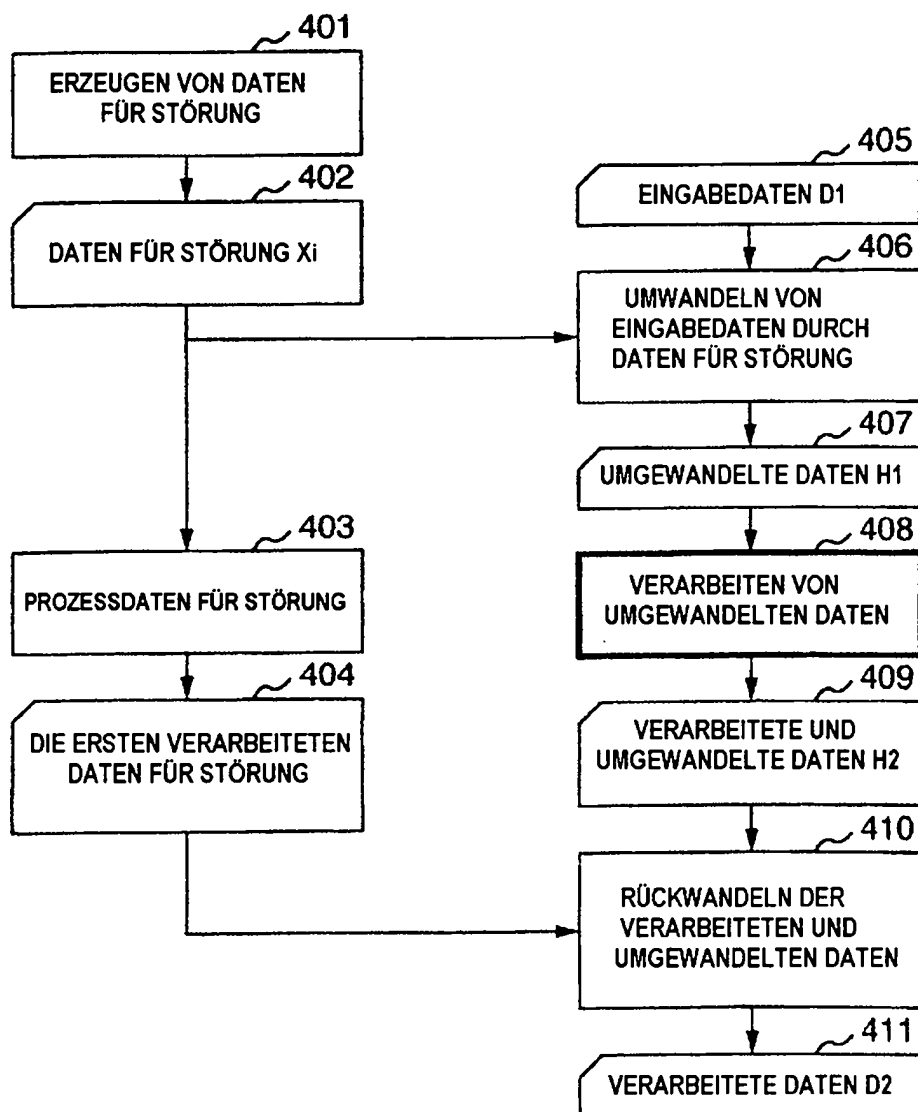


FIG. 5

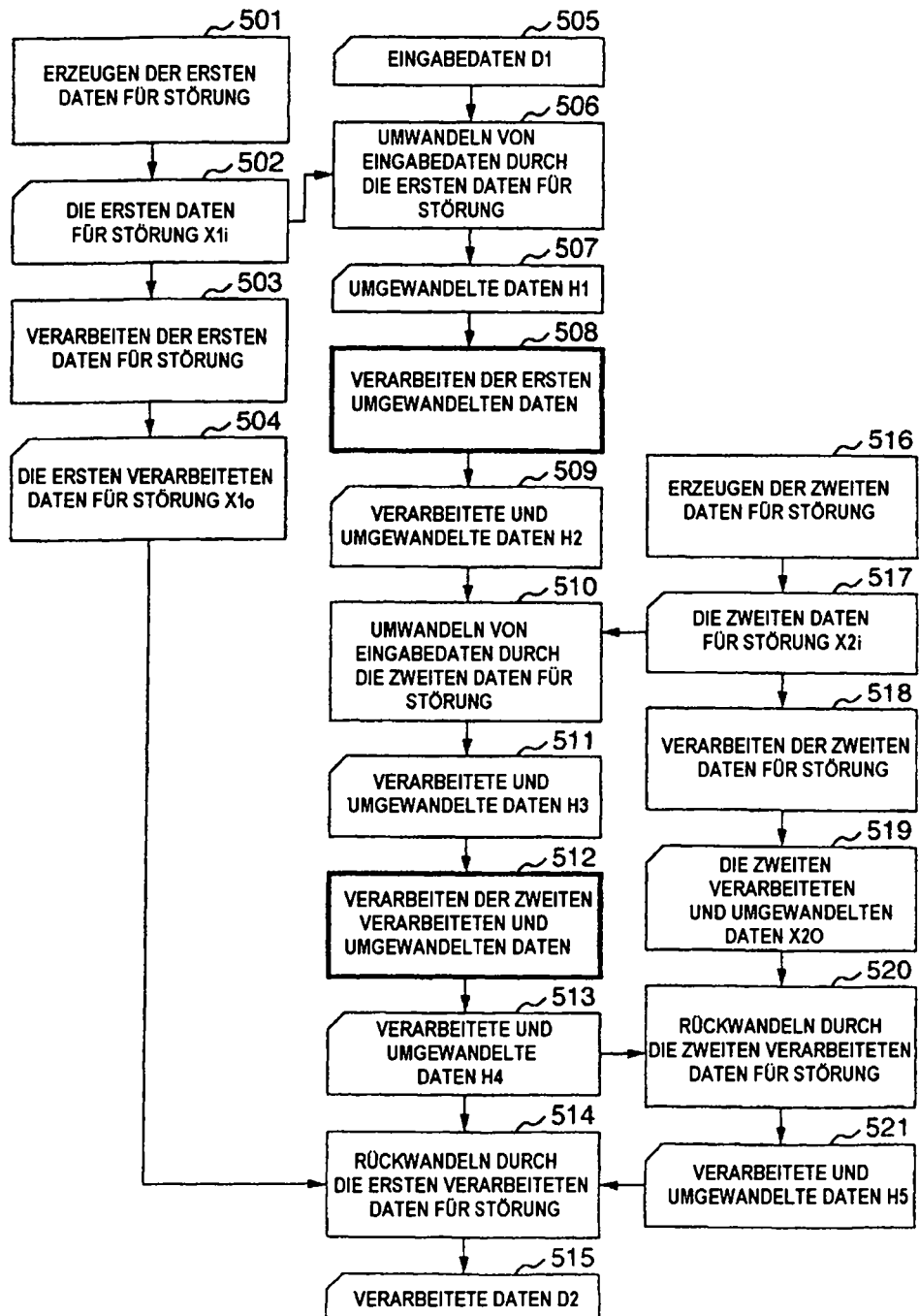


FIG. 6

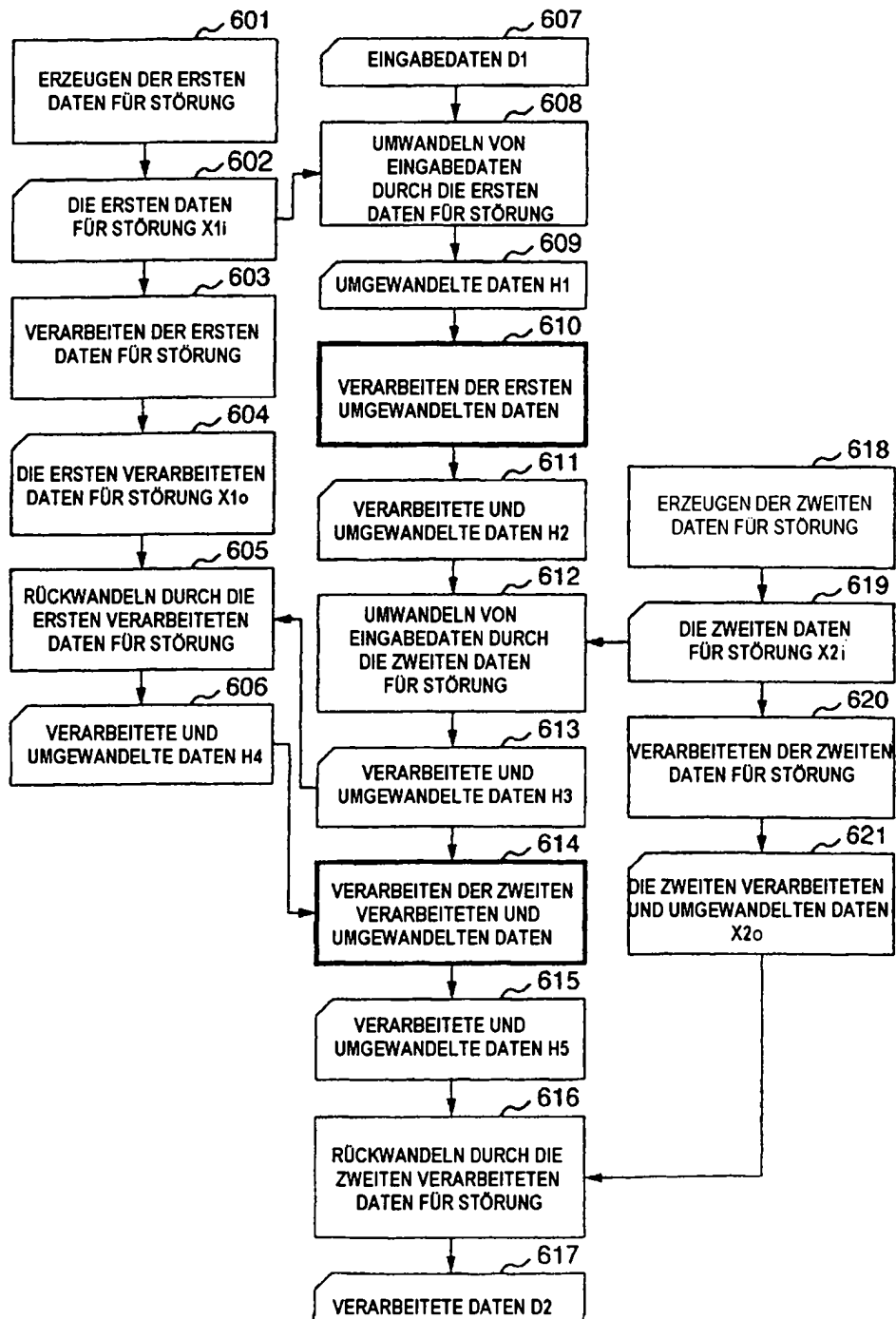


FIG. 7

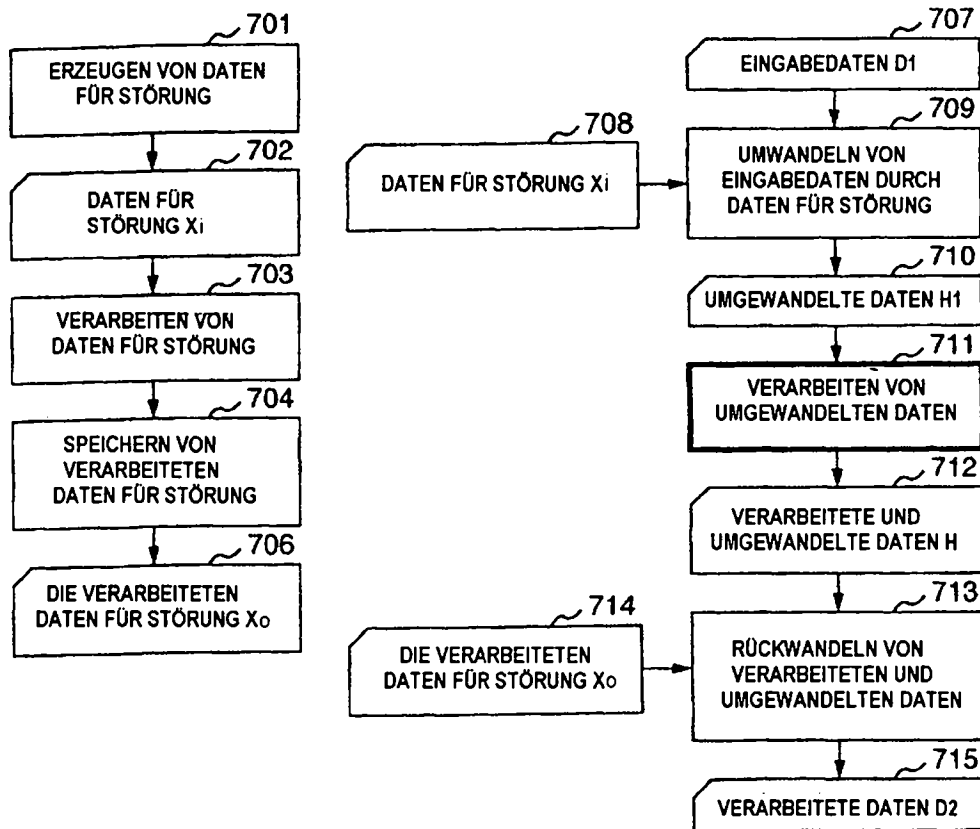


FIG. 8

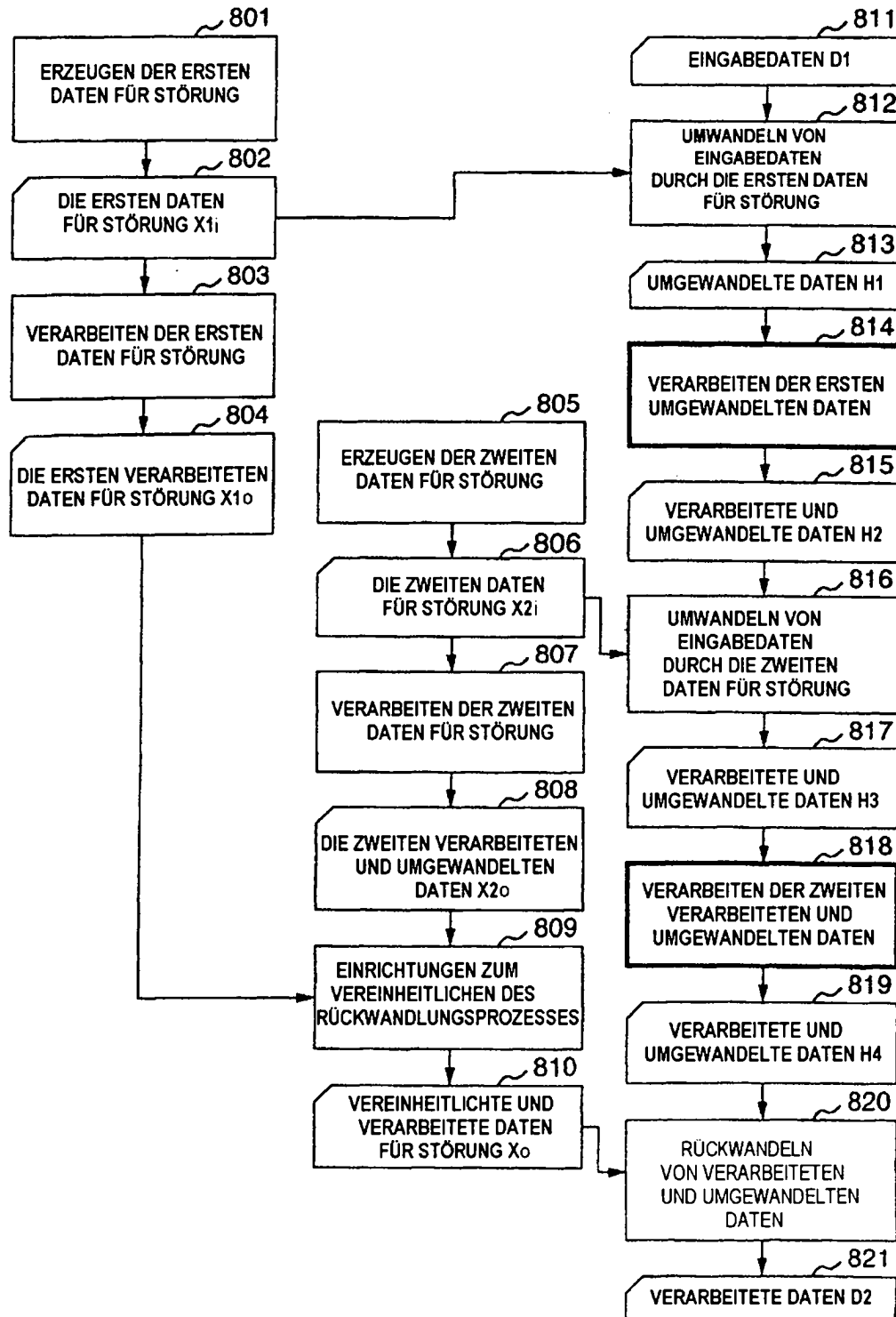


FIG. 9

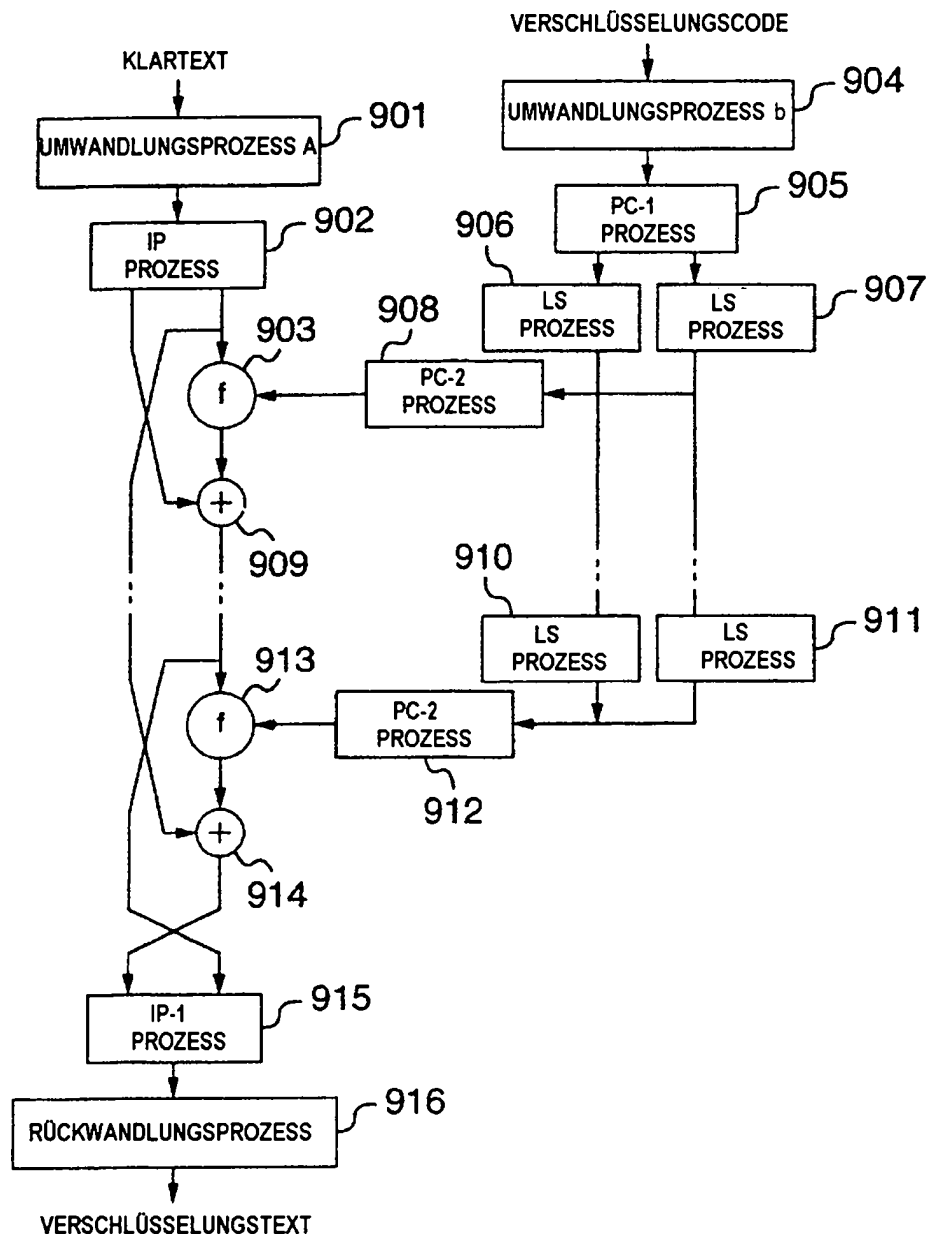


FIG. 10

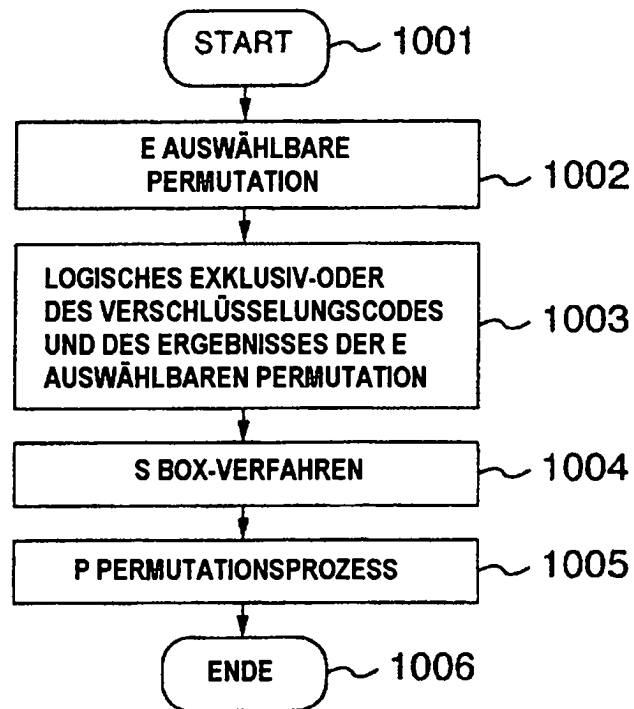


FIG. 11

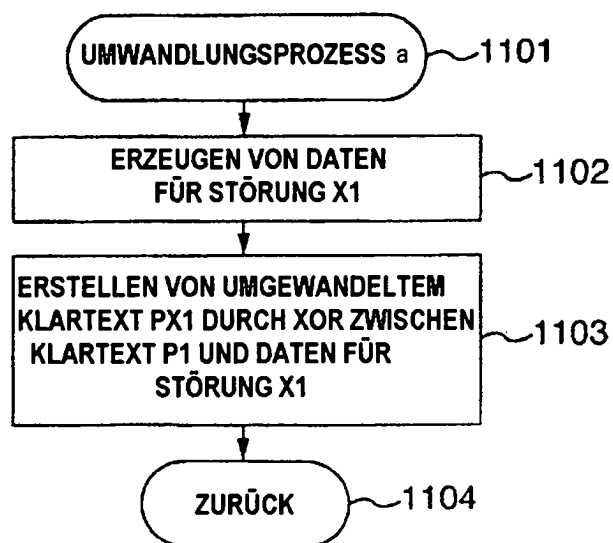


FIG. 12

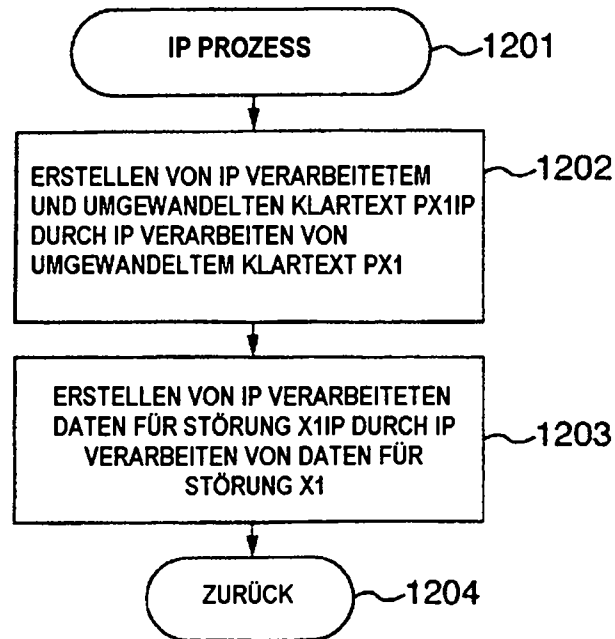


FIG. 13

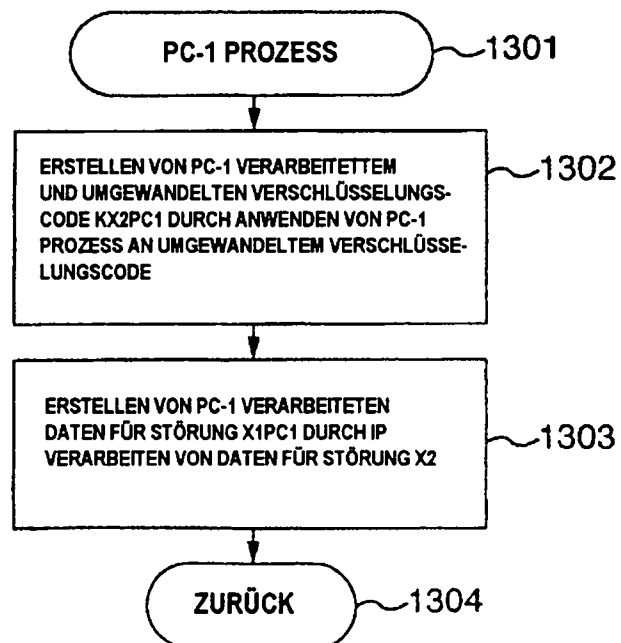


FIG. 14

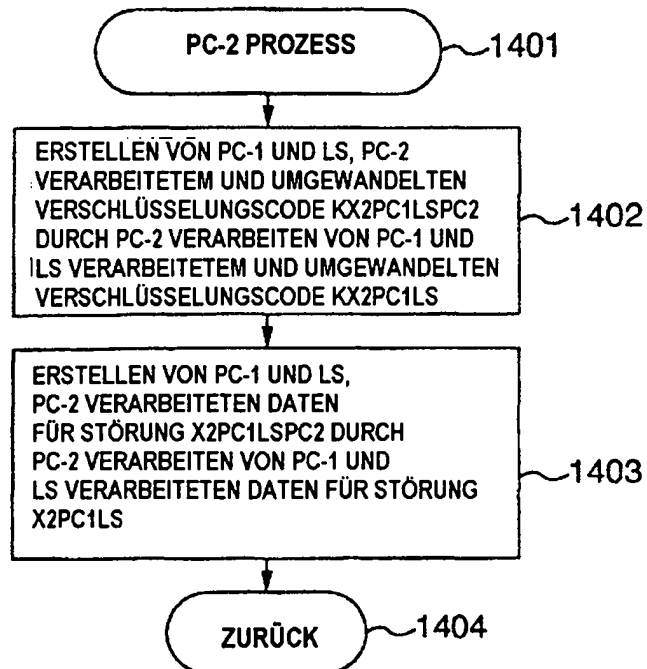


FIG. 15

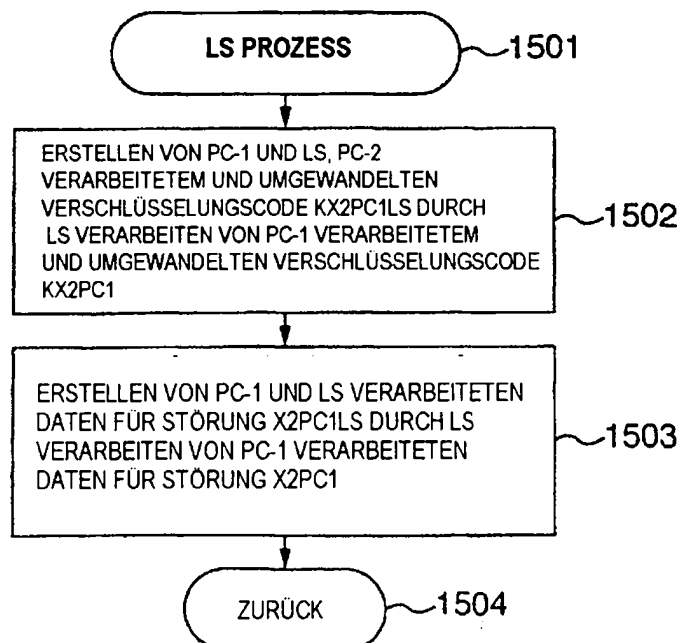


FIG. 16

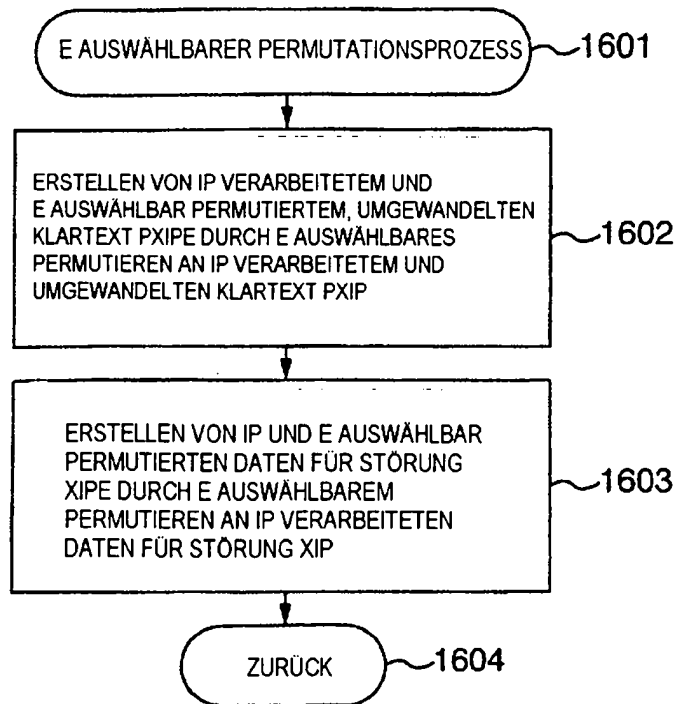


FIG. 17

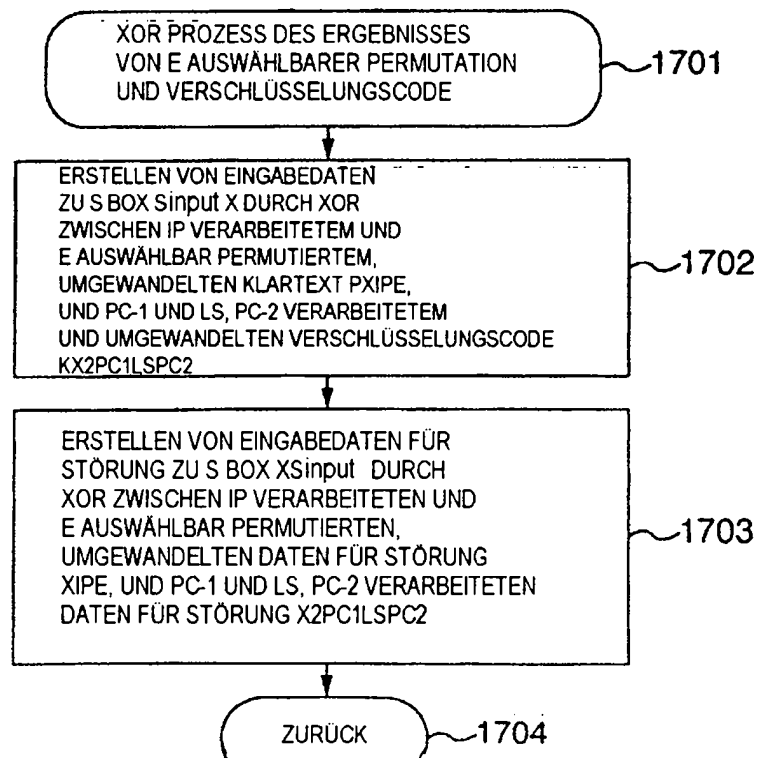


FIG. 18

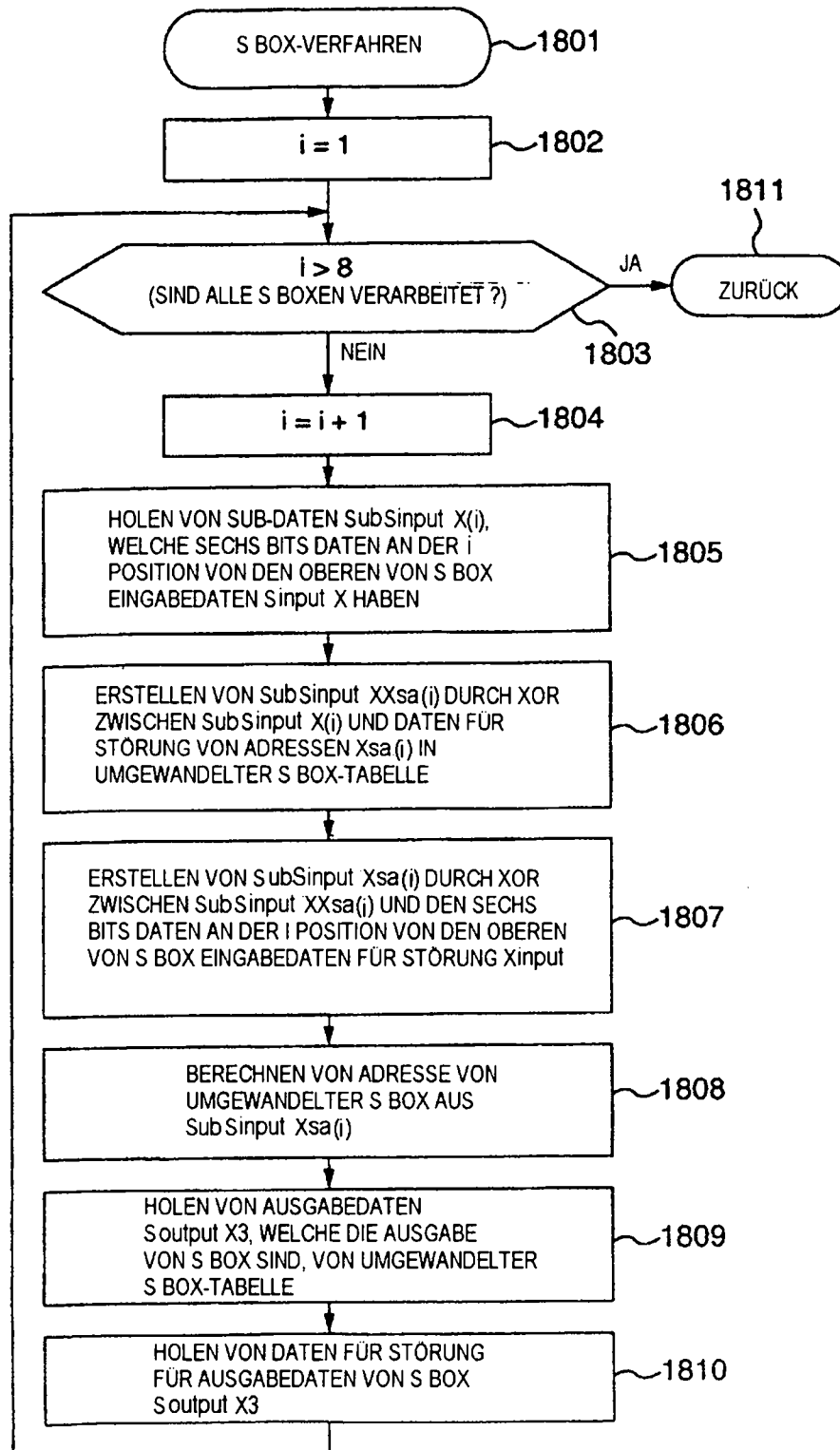


FIG. 19

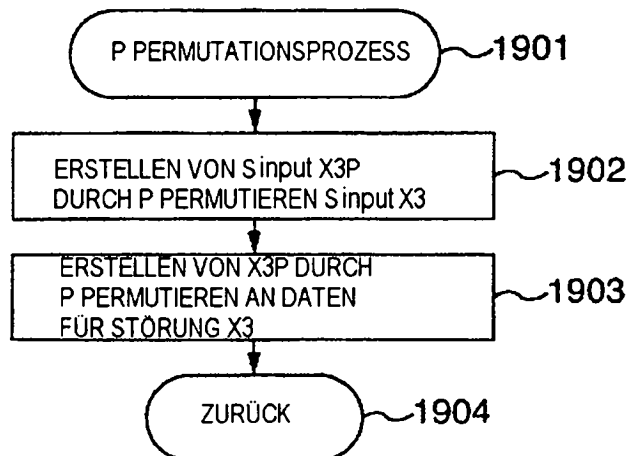


FIG. 20

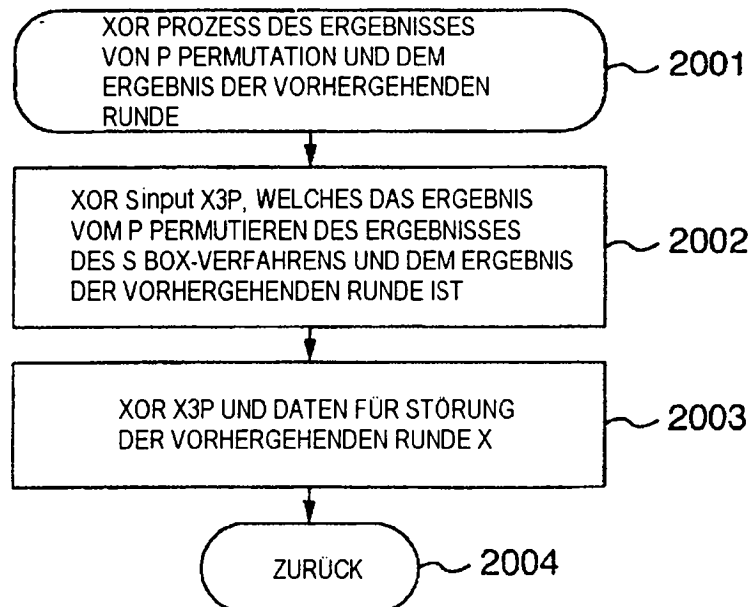


FIG. 21

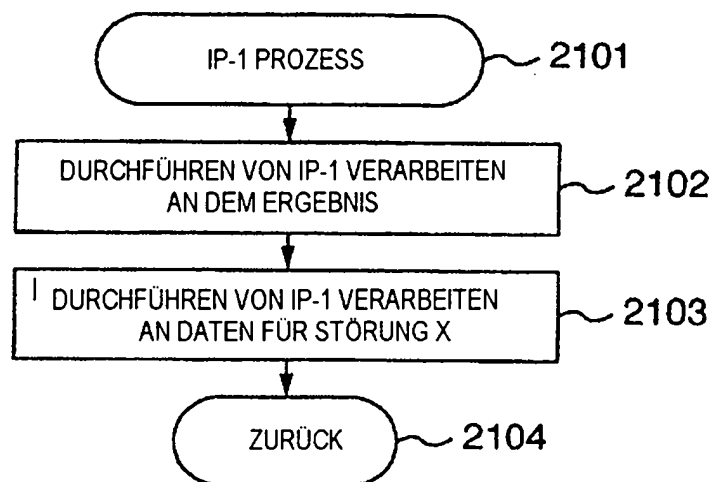


FIG. 22

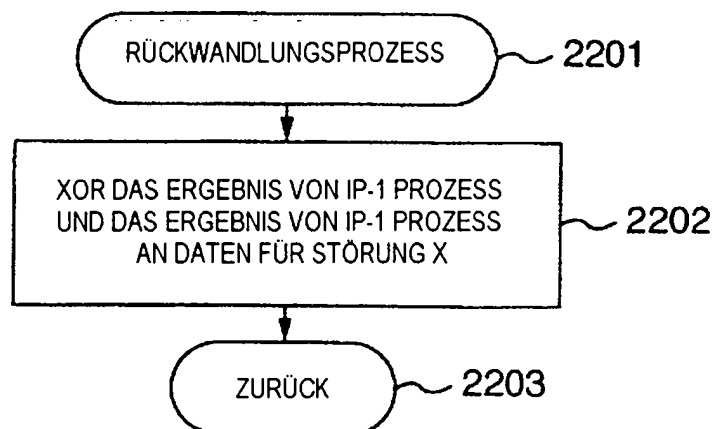


FIG. 23

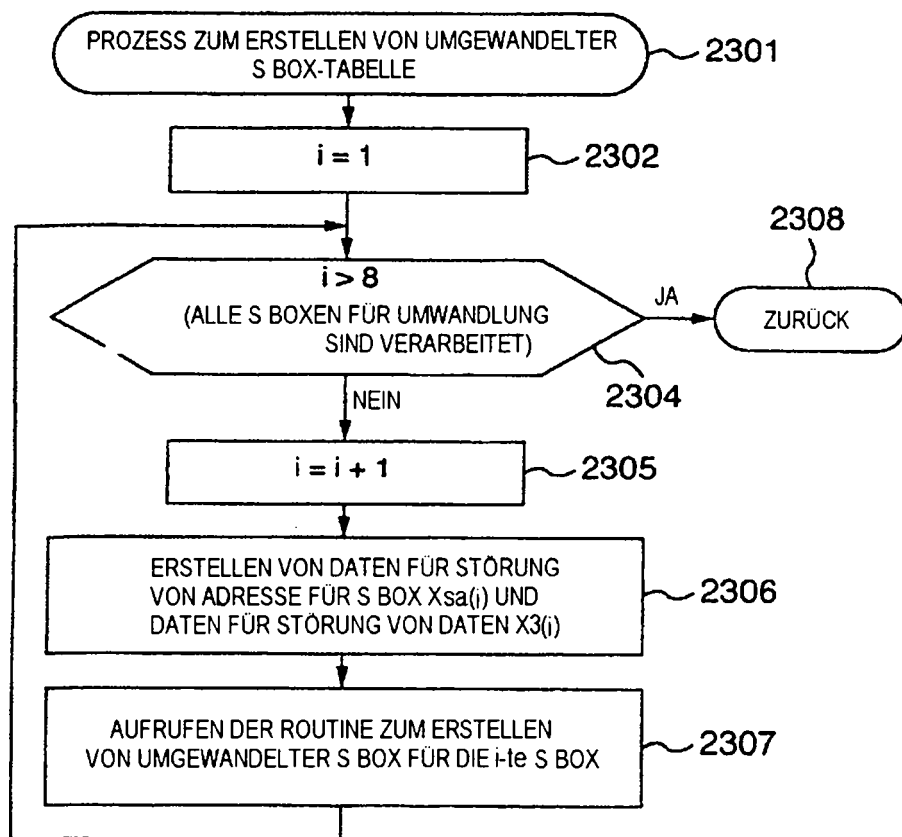


FIG. 24

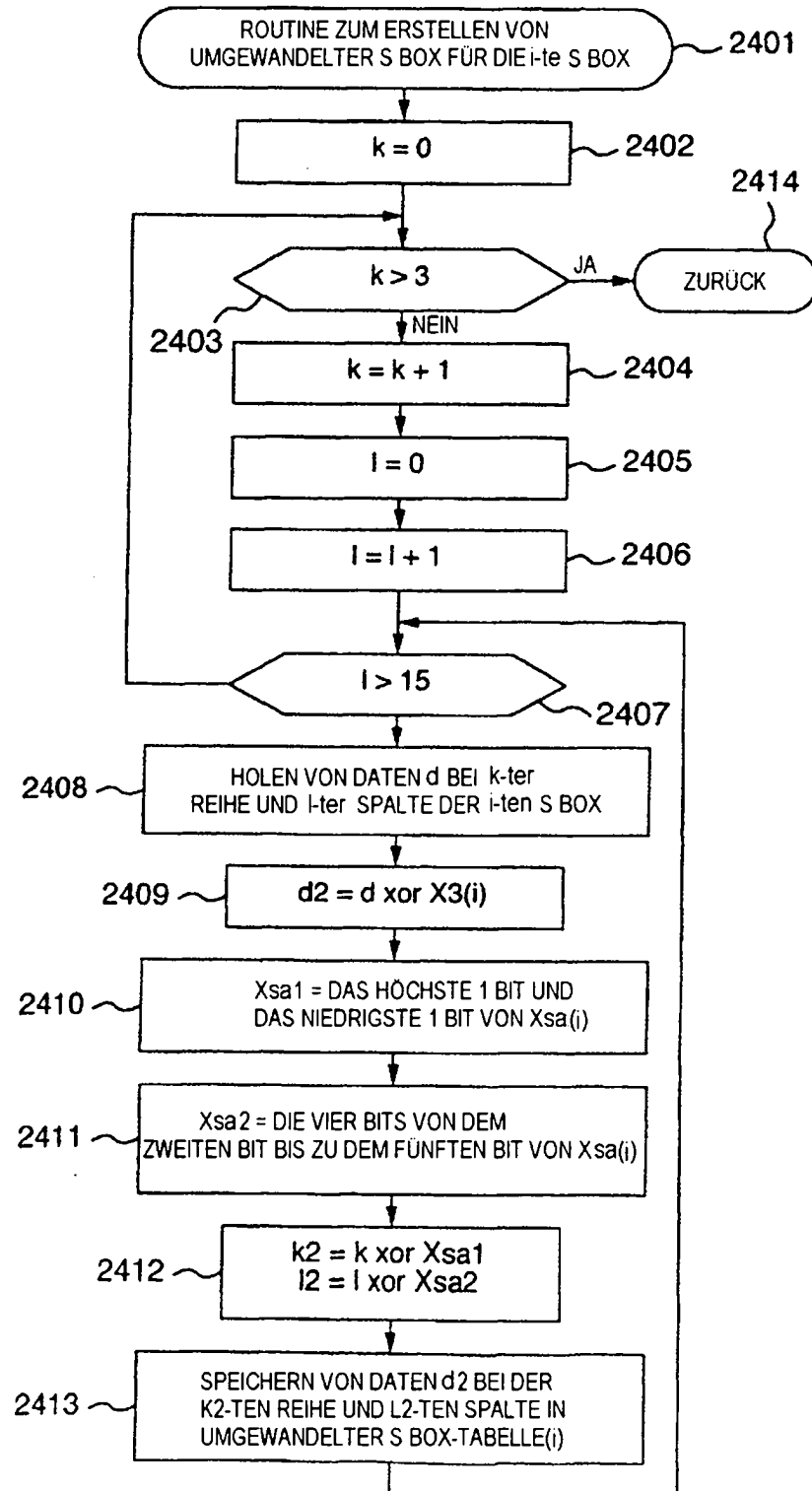


FIG. 25

2501

	0	1	2	3		15
0	14	4	13	1		7
1	0	15	7	4		8
2	4	1	14	8		0
3	15	12	8	2		13

2502 2503 2504

FIG. 26

2601 2606

	0	1	2	3		15
0	9	3	10	6		0
1	7	8	0	3		15
2	3	6	9	15		7
3	8	11	15	5		10

2602 2603 2604 2605

XOR 7 AN DER TABELLE IN FIGUR 25

FIG. 27

2701		2706		2703	
	9	8	11	20	6
2	9	3	10	6	0
3	7	8	0	3	15
0	3	6	9	15	7
1	8	11	15	5	10
2702		2704			

XOR 2 UND 9 AN JEWEILS DER REIHE
UND SPALTE IN DER TABELLE IN FIGUR 26

FIG. 28

	1	2	3	4	5	6
1	32	1	2	3	4	5
2	4	5	6	7	8	9
3	8	9	10	11	12	13
4	12	13	14	15	16	17
5	16	17	18	19	20	21
6	20	21	22	23	24	25
7	24	25	26	27	28	29
8	28	29	30	31	32	1

FIG. 29

16	7	20	21	2901
29	12	28	17	
1	15	23	26	
5	18	31	10	
2	8	21	14	
32	27	3	9	
19	13	30	6	
22	11	4	25	

FIG. 30

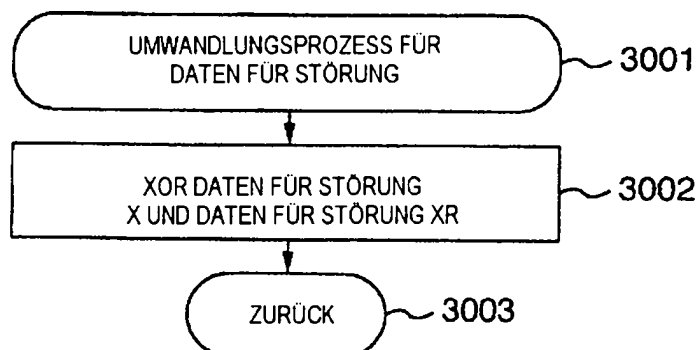


FIG. 31

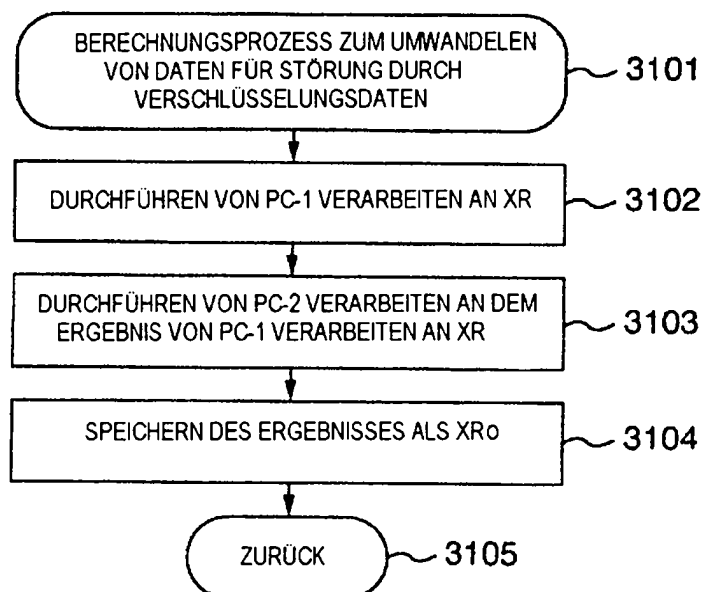


FIG. 32

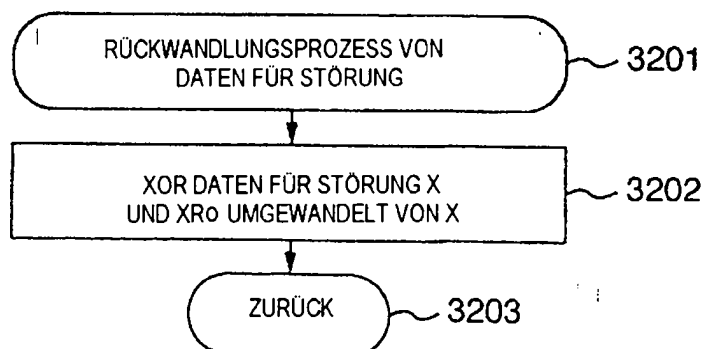


FIG. 33

3301

	0	1	2	3
0	B	3	8	7
1	1	4	0	D
2	2	F	A	9
3	6	C	5	E

FIG. 34

3401

	0	1	2	3
0	2	A	1	E
1	8	D	9	4
2	B	6	3	0
3	F	5	C	7

FIG. 35

	0	1	2	3
0	5	F	7	C
1	6	B	0	3
2	D	8	4	9
3	A	2	E	1

3501

FIG. 36

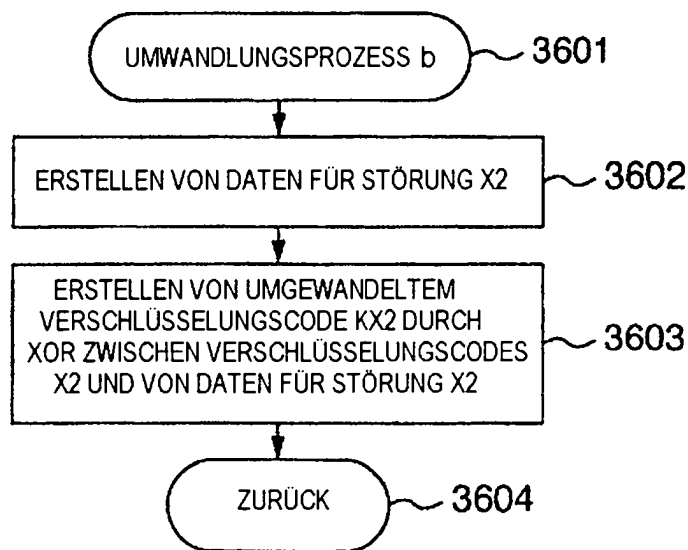


FIG. 37

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

FIG. 38

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4