(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 695 34 603 T2** 2006.08.03

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 804 758 B1**
(21) Deutsches Aktenzeichen: **695 34 603.2**
(86) PCT-Aktenzeichen: **PCT/CA95/00452**
(96) Europäisches Aktenzeichen: **95 926 348.4**
(87) PCT-Veröffentlichungs-Nr.: **WO 1996/004602**
(86) PCT-Anmeldetag: **31.07.1995**
(87) Veröffentlichungstag der PCT-Anmeldung: **15.02.1996**
(97) Erstveröffentlichung durch das EPA: **05.11.1997**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **09.11.2005**
(47) Veröffentlichungstag im Patentblatt: **03.08.2006**

(51) Int Cl.[8]: ***G06F 7/72*** (2006.01)

(30) Unionspriorität:
282263 **29.07.1994** **US**

(84) Benannte Vertragsstaaten:
CH, DE, FR, GB, LI

(73) Patentinhaber:
Certicom Corp., Mississauga, Ontario, CA

(74) Vertreter:
Klunker, Schmitt-Nilson, Hirsch, 80797 München

(72) Erfinder:
MULLIN, C., Ronald, Waterloo, CA; VANSTONE, A., Scott, Waterloo, CA; AGNEW, B., Gordon, Campbellville, CA

(54) Bezeichnung: **VERSCHLÜSSELUNGSSYSTEM FÜR ELLIPTISCHE KURVE**

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

GEBIET DER ERFINDUNG

[0001] Die Erfindung betrifft die Verschlüsselung mit öffentlichem Schlüssel.

[0002] Die zunehmende Verwendung und Weiterentwicklung der Datenübertragung auf den Gebieten wie Telekommunikation, Netzwerkbetrieb, Zellulare Kommunikation, Funk-Kommunikation, "Smart Card"-Anwendungen, audiovisuelle und Video-Kommunikationen hat zu einem zunehmenden Bedarf an Systemen geführt, die eine Datenverschlüsselung, Authentifizierung und Verifizierung gestatten.

[0003] Es ist bekannt, dass Daten unter Verwendung eines Schlüsselpaars verschlüsselbar sind, wobei der eine Schlüssel öffentlich und der andere privat ist. Die Schlüssel stehen mathematisch derart in Beziehung, dass mit dem öffentlichen Schlüssel verschlüsselte Daten nur mit dem privaten Schlüssel entschlüsselt werden können, und umgekehrt, wobei mit dem privaten Schlüssel verschlüsselte Daten nur mit dem öffentlichen Schlüssel entschlüsselt werden können. Auf diese Weise kann der öffentliche Schlüssel eines Empfängers derart verfügbar gemacht werden, dass für den Empfänger vorgesehene Daten mit dem öffentlichen Schlüssel verschlüsselt werden können und nur mit dem privaten Schlüssel des Empfängers entschlüsselt werden können, oder umgekehrt, verschlüsselte gesendete Daten als authentisch verifiziert sind, wenn sie mit dem öffentlichen Schlüssel des Senders entschlüsselt werden.

[0004] Es ist bekannt, dass durch die Verwendung von Berechnungen in einem endlichen Körper, dessen Elemente auf einer elliptischen Kurve liegen, d.h. durch Definieren einer Gruppenstruktur G auf den Lösungen von $y^2 + xy = x^3 + ax^2 + b$ über einem endlichen Körper, das Problem aufgrund der Attribute elliptischer Kurven schwierig ist. Deshalb ist es möglich, ein höheres Sicherheitsmaß für eine gegebene Schlüsselgröße zu erreichen. Alternativ kann ein verkleinerter Schlüssel dazu benutzt werden, ein gewünschtes Maß an Sicherheit zu bewahren.

[0005] Die durch den Gebrauch elliptischer Kurven geschaffene inhärente Sicherheit wird abgeleitet aus der Besonderheit, dass die Addition zweier Punkte auf der Kurve definiert werden kann als weiterer Punkt, der sich auf der Kurve befindet. In ähnlicher Weise resultiert das Ergebnis der Addition eines Punkts mit sich selbst zu einem weiteren, auf der Kurve liegenden Punkt. Durch Auswählen eines Anfangspunkts auf der Kurve und Multiplizieren des Punkts mit einer ganzen Zahl wird also ein neuer Punkt gewonnen, der auf der Kurve liegt. Dies bedeutet, dass wenn P = (x, y) ein Punkt auf einer elliptischen Kurve über einem endlichen Körper $[E(F_q^n)]$ mit x und y jeweils als Vektor aus n Elementen ist, dann von einem weiteren Punkt $R \in < P >$ (die von P erzeugte Untergruppe) gilt dP = R. Um ein solches Schema anzugehen, besteht die Aufgabe darin, ein effizientes Verfahren zum Auffinden einer ganzen Zahl d, $0 \leq d \leq$ (Grad von P)–1 derart aufzufinden, dass dP = R. Um ein solches Schema aufzubrechen, besitzen die bislang besten bekannten Algorithmen Laufzeiten von nicht besser als $O(\sqrt{P})$, mit p als größte Teiler-Primzahl für den Grad der Kurve (die Anzahl von Punkten auf der Kurve).

[0006] Damit lässt sich in einem kryptografischen System, in welchem die ganze Zahl d geheim bleibt, die Schwierigkeit ausnutzen, um d zu bestimmen.

[0007] Aus Menezes A J et al. "Elliptic Curve Cryptosystems And Their Implementation", Journal of Cryptology, vol. 6, no. 4, 1. Januar 1992 (1992-01-01), Seiten 209–224, XP002069135 ist es bekannt, Koordinaten eines auf einer elliptischen Kurve liegenden Punkt dadurch zu übermitteln, dass eine x-Koordinate und ein einzelnes Bit der y-Koordinate gesendet werden, woraus die vollständige y-Koordinate durch Berechnungen gewonnen werden kann.

[0008] Aspekte der Erfindung sind in den beigefügten Ansprüchen angegeben.

[0009] Im Folgenden wird eine Ausführungsform der Erfindung beispielhaft unter Bezugnahme auf die begleitenden Zeichnungen beschrieben. Es zeigen:

[0010] Fig. 1 ein Diagramm der Übertragung einer verschlüsselten Nachricht von einem Ort zu einem anderen,

[0011] Fig. 2 ein Diagramm eines Verschlüsselungsmoduls, welches in dem in Fig. 1 gezeigten Kommunikationssystem verwendet wird,

[0012] Fig. 3 ein Diagramm eines bei dem Verschlüsselungs- und Entschlüsselungsmodul nach Fig. 2 verwendeten Prozessors für endliche Körper.

[0013] Fig. 4 ist ein Flussdiagramm, welches die Bewegung der Elemente durch den Prozessor nach Fig. 3 bei der Berechnung einer inversen Funktion veranschaulicht.

[0014] Fig. 5 ist ein Flussdiagramm, welches die Bewegung der Elemente durch den Prozessor nach Fig. 3 bei der Berechnung der Addition zweier Punkte veranschaulicht.

[0015] Im Folgenden wird eine Ausführungsform der Erfindung anhand eines ElGamal-Schlüsselaustauschprotokolls und eines Galois-Felds F_2^{155} zum Erläutern der grundlegenden Prinzipien erläutert. Anschließend werden Verfeinerungen beschrieben.

Systemkomponenten

[0016] Nach Fig. 1 soll eine Nachricht M von einem Sender **2** über einen Übertragungskanal **14** zu einem Empfänger **12** übertragen werden. Jeder Sender **10** und Empfänger **12** besitzt ein ihm zugehöriges Verschlüsselungs-/Entschlüsselungsmodul **16** zum Implementieren eines Schlüsselaustauschprotokolls und eines Verschlüsselungs-/Entschlüsselungsalgorithmus.

[0017] Das Modul ist **16** schematisch in Fig. 2 dargestellt und enthält eine arithmetische Einheit **20** zum Durchführen von Berechnungen beim Schlüsselaustausch und bei der Schlüsselerzeugung. Ein Privatschlüsselregister **22** enthält einen Privatschlüssel d, erzeugt als 155 Bit lange Datenkette durch einen Zufallszahlgenerator **24** und verwendet zum Erzeugen eines öffentlichen Schlüssels, der in einem Register **26** für öffentliche Schlüssel gespeichert wird. Ein Basispunktregister **28** enthält die Koordinaten eines Basispunkts P, der in der ausgewählten elliptischen Kurve mit jeder Koordinate (x, y) liegt, dargestellt als 155 Bit lange Datenkette.

[0018] Jede der Datenketten ist ein Vektor aus Binärziffern, wobei jede Ziffer der Koeffizient eines Elements des endlichen Körpers in der Normalbasisdarstellung der Koordinate ist.

[0019] Die ausgewählte elliptische Kurve hat die allgemeine Form $y^2 + xy = x^3 + ax^2 + b$, wobei die Parameter dieser Kurve, nämlich die Koeffizienten a und b, in einem Parameterregister **30** gespeichert sind. Die Inhalte der Register **22**, **24**, **26**, **28** und **30** können unter der Steuerung einer CPU **32** bedarfsweise zu der arithmetischen Einheit **20** transferiert werden.

[0020] Die Inhalte des Registers für öffentliche Schlüssel **26** stehen auch für den Übertragungskanal **14** bei Empfang einer entsprechenden Anforderung zur Verfügung. In der einfachsten Implementierung arbeitet jedes Verschlüsselungsmodul **16** in einer gemeinsamen Sicherheitszone mit derselben Kurve und demselben Basispunkt, so dass die Inhalte der Register **28** und **30** nicht zugänglich sein müssen. Wenn eine weitere Verfeinerung erforderlich ist, kann allerdings jedes Modul **16** seine eigene Kurve und eigenen Basispunkt auswählen, in welchem Fall die Inhalte der Register **28** und **30** für den Kanal **14** zugänglich sein müssen.

[0021] Das Modul **16** enthält weiterhin ein Ganzzahlregister **34**, welches eine ganze Zahl k, den "Session-Seed", von dem Generator **24** zur Verwendung bei der Verschlüsselungen und beim Schlüsselaustausch aufnimmt. Das Modul **16** besitzt einen Schreib-/Lesespeicher (RAM) **36**, der als Zwischenspeicher während Berechnungen fungiert.

[0022] Das Verschlüsseln der Nachricht M mit einem Chiffrierschlüssel kdP, der von dem öffentlichen Schlüssel dP und der Session-Seed-Ganzzahl k abgeleitet ist, erfolgt in einer Verschlüsselungseinheit **40**, die einen ausgewählten Verschlüsselungsalgorithmus implementiert. Ein einfacher wenngleich effektiver Algorithmus besteht in einer XOR-Funktion, welche die Nachricht m mit den 310 Bits des Schlüssels kdP einer Exklusiv-Oder-Verknüpfung unterzieht.

[0023] Ein alternatives Verschlüsselungsprotokoll behandelt die Nachricht m_1, m_2, jeweils mit einer Länge von 155 Bits im Fall von F_2^{155}, und bildet eine XOR-Verknüpfung der Nachricht m_1, m_2 mit den Koordinaten des Session-Schlüssels kdP, um ein Paar Bit-Ketten $(m_1 \oplus x_0)$ $(m_2 \oplus y_0)$ zu bilden. Zur weiteren Sicherheit wird außerdem ein Paar Körperelemente z_1z_2 aus den Koordinaten (x_0y_0) des kdP gebildet.

[0024] In einer Ausführungsform werden die Elemente z_1z_2 aus der Aneinanderreihung eines Teils von x_0 mit einem Teil von y_0 gebildet, beispielsweise $z_1 = x_{01}||y_{02}$ und $z_2 = x_{02}||y_{01}$

wobei x_{01} die erste Hälfte der Bit-Kette von x_0 ist
x_{02} die zweite Hälfte der Bit-Kette von x_0 ist
y_{01} die erste Hälfte der Bit-Kette von y_0 ist
y_{02} die zweite Hälfte der Bit-Kette von y_0 ist.

[0025] Die ersten Elemente z_1 und z_2, die als Körperelemente behandelt werden, werden anschließend mit den jeweiligen Bit-Ketten $(m_1 \oplus x_0)$ und $(m_2 \oplus y_0)$ multipliziert, um die Bit-Ketten c_1c_2 des chiffrierten Textes c zu bilden. d.h.

$c_1 = z_1 (m_1 \oplus x_0)$

$C_2 = z_2 (m_2 \oplus y_0)$

[0026] In einer bevorzugten Implementierung des Verschlüsselungsprotokolls wird anstelle von y_0 in der obigen Ausführungsform eine Funktion von x_0 verwendet, beispielsweise dient die Funktion x_0^3 als die zweite 155-Bit-Kette, so dass

$c_1 = z_1 (m_1 \oplus x_0)$

$c_2 = z_2(m_2 \oplus x_0^3)$

und

$Z_1 = X_{01}||x_{02}^3$

$z_2 = X_{02}||x_{01}^3$

wobei x_{01}^3 die erste Hälfte von x_0^3 ist und
x_{02}^3 die zweite Hälfte von x_0^3 ist.

[0027] Dieses Protokoll ist auch anwendbar auf die Implementierung einer Verschlüsselung mit elliptischer Kurve in einem anderen Körper als F_2^m beispielsweise Z_P oder allgemein F_p^m.

[0028] Wenn Z_P verwendet wird, kann es notwendig sein, die Werte von x_0 und y_0 oder x_0^3 einzustellen, um einen Überlauf bei der Multiplikation mit z_1 und z_2 zu vermeiden. Üblicherweise geschieht dies durch Einstellen des höchstwertigen Bits x_0 und F_p^m oder y_0 auf null.

Schlüsselerzeugung, -austausch und Verschlüsselung

[0029] Damit der Sender **10** die Nachricht M an den Empfänger **12** sendet, wird der öffentliche Schlüssel des Empfängers von dem Sender **10** erhalten. Der öffentliche Schlüssel wird von dem Empfänger **12** dadurch erhalten, dass er das Produkt des geheimen Schlüssels d und des Basispunkts P in der arithmetischen Einheit **20** berechnet, wie im Folgenden ausführlich erläutert wird. Das Produkt dP stellt einen Punkt auf der ausgewählten Kurve dar und dient als öffentlicher Schlüssel. Der öffentliche Schlüssel dP wird in Form von zwei 155 Bit langen Datenketten in dem Register **26** für öffentliche Schlüssel gespeichert.

[0030] Nach Erhalt des öffentlichen Schlüssels dP durch den Sender **10** wird der Schlüssel im RAM **36** gespeichert. Man sieht, dass selbst dann, wenn der Basispunkt P bekannt und öffentlich verfügbar ist, die Attribute der elliptischen Kurve das Herleiten des geheimen Schlüssels d verhindern.

[0031] Der Sender **10** verwendet die arithmetische Einheit **20** zum Berechnen des Produkts des Session-Seeds k und des öffentlichen Schlüssels dP und speichert das Ergebnis kdP in dem RAM **36** zur Verwendung bei dem Verschlüsselungs-Algorithmus. Das Ergebnis kdP ist ein weiterer Punkt auf der ausgewählten Kurve, wiederum dargestellt durch zwei 155 Bit lange Datenketten oder Vektoren, und dient als Chiffrierschlüssel.

[0032] Der Sender **10** berechnet außerdem das Produkt des Session-Seeds k und des Basispunkts P, um einen neuen Punkt kP, nämlich den öffentlichen "Session-Schlüssel" oder "Sitzungs-Schlüssel" bereitzustellen, der in dem RAM **36** gespeichert wird.

[0033] Der Sender **10** ist nun im Besitz des öffentlichen Schlüssels dP des Empfängers **12**, eines öffentlichen Session-Schlüssels kP und eines Chiffrierschlüssels kdP und kann diese dazu benutzen, eine verschlüsselte, d.h. chiffrierte Nachricht zu senden. Der Sender **10** verschlüsselt die Nachricht M mit dem Chiffrierschlüssel kdP in der Verschlüsselungseinheit **40**, wobei die ausgewählten, oben diskutierten Verschlüsselungsprotokolle implementiert werden, um eine verschlüsselte Nachricht C zu bilden. Der chiffrierte Text C wird zusammen mit dem Wert kP an das zu dem Empfänger **12** gehörige Verschlüsselungsmodul **16** gesendet.

[0034] Der Empfänger **12** macht Gebrauch von dem öffentlichen Session-Schlüssel kP mit Hilfe seines Privatschlüssels d, um den Chiffrierschlüssel kdP in der Arithmetikeinheit **20** zu berechnen und anschließend den chiffrierten Text C in der Verschlüsselungseinheit **40** zu entschlüsseln und die Nachricht M wiederzugewinnen.

[0035] Während dieses Austauschvorgangs bleiben der geheime Schlüssel d und der Session-Seed k geheim und sicher. Obwohl P, kP und dP bekannt sind, lässt sich der Chiffrierschlüssel kdP nicht berechnen, bedingt durch die Schwierigkeit des Erhaltens von entweder d oder k.

[0036] Die Effizienz der Verschlüsselung hängt ab von der effizienten Berechnung der Werte kP, dP und kdP mit Hilfe der arithmetischen Einheit **20**. Jede Berechnung erfordert die wiederholte Addition von zwei Punkten auf der Kurve, was wiederum die Berechnung von Quadraten und Inversen in F_2^m erfordert.

Arbeitsweise der arithmetischen Einheit

[0037] Der Betrieb der arithmetischen Einheit **20** ist schematisch in **Fig. 3** dargestellt. Die Einheit **20** enthält einen Multiplizierer **48** mit einem Paar zyklischer Schieberegister **42**, **44** und einem Akkumulatorregister **46**. Jedes der Register **42**, **44** und **46** enthält M Zellen **50a**, **50b**...**50m**, im vorliegenden Beispiel 155 Zellen, um die m Elemente einer Normalbasisdarstellung einer der Koordinaten von beispielsweise x von P aufzunehmen. Wie vollständig in dem US-Patent 4 745 568 erläutert ist, sind die Zellen **50** der Register **42**, **44** mit entsprechenden Zellen **50** des Akkumulatorregisters **46** derart verbunden, dass in jeder Zelle des Registers **46** ein gruppenweiser Term erzeugt wird. Die Register **42**, **44** und **46** sind außerdem direkt bitweise verbunden, um rasche Transfers von Daten zwischen den Registern zu ermöglichen.

[0038] Die Bewegung der Daten durch die Register wird gesteuert von einem Steuerregister **52**, welches den Befehlssatz gemäß folgender Tabelle ausführen kann:

TABELLE 1

BEFEHLSSATZ

Operation	Größe	Taktzyklen
Körpermultiplikation MULT	155-Bit-Blöcke	156
Berechnen der Inversen INVERSE	24 Multiplikationen	etwa 3800
I/O	5-32 Bit-Transfers pro 10 Taktzyklen	10
WRITE (A, B oder C) READ (A, B oder C)	Lesen/Schreiben in Register	2 Taktzyklen pro Transfer
Elementarregister	155-Bit-Paralleloperation	
(leer) NOP		
Rotate (A, B oder C)		
Copy		
(A←B)		
(A←C)		
(A←B)		
(B←C)		
SWAP (A↔B)		
CLEAR (A, B oder C)		
SET (A, B oder C)		
ADD (A⊕B)		
ACCUMULATE		

[0039] Die Einheit **20** enthält einen Addierer **54** zum Empfangen von Daten aus den Registern **42**, **44**, **46** und dem RAM **36**. Der Addierer **54** bildet eine XOR-Funktion, sein Ausgangssignal entspricht einem Datenstrom, der in dem RAM **36** oder in einem der Register **42**, **44** gespeichert werden kann. Obwohl als serielles Bauelement dargestellt, ist ersichtlich, dass er in Form zweier paralleler Bauelemente implementiert werden kann, um die Berechnungszeit zu verbessern. In ähnlicher Weise können die Register **42**, **44** und **46** parallel geladen werden. Jedes der Register **42**, **44** und **46** ist ein 155 Bit langes Register und wird adressiert von einem 32-Bit-Datenbus, so dass 32 Bits Daten in zwei Taktzyklen transferiert werden können und der gesamte Ladevorgang in fünf Operationen geschieht.

[0040] Die bei der Berechnung verwendeten Unterroutinen werden im Folgenden diskutiert.

a) Multiplikation

[0041] Das zyklische Verschieben der Elemente durch die Register **42** und **44** mit m Wiederholungen in Verbindung mit einer zugehörigen Verschiebung des Akkumulatorregisters **46** akkumuliert aufeinander folgende Gruppenterme in einzelnen Akkumulatorzellen, wobei eine vollständige Umwälzung der Elemente in den Registern **42** und **44** die Elemente des Produkts im Akkumulatorregister **46** bildet.

b) Quadrierung

[0042] Durch Arbeiten in F_2^m und Einführen einer Normalbasisdarstellung der Körperelemente kann der Multiplizierer **48** auch das Quadrat einer Zahl dadurch bilden, dass die Elemente einer Zelle durch das Register **42** zyklisch verschoben werden. Nach einer Zellenverschiebung repräsentieren die in dem Register enthaltenen Elemente das Quadrat der Zahl. Allgemein lässt sich eine Zahl zur Potenz **29** erheben, indem man g-mal eine zyklische Verschiebung durch ein Register vollzieht.

c) Inversion

[0043] Die Berechnung der Inversen einer Zahl lässt sich in effizienter Weise mit dem Multiplizierer **48** dadurch vornehmen, dass man einen Algorithmus implementiert, der von mehreren Quadrierungen Gebrauch macht. Die Inverse X^{-1} wird dargestellt in der Form

$$X^{2^M-2} \text{ oder } X^{2(2^{M-1}-1)}.$$

.

[0044] Wenn man m-1 als das Produkt zweier Faktoren g, h ansieht, lässt sich X^{-1} umschreiben in der Form

$$X^{2(2^{gh}-1)}$$

der

$$\beta^{2^{gh}-1}$$

, mit $\beta = X^2$.

[0045] Der Exponent $2^{gh}-1$ ist äquivalent zu

$$(2^g-1)\left(\sum_{i=0}^{h-1} 2^{ig}\right)$$

[0046] Der Term 2^g-1lässt sich schreiben in der Form

$$X^{-1} = \beta^{\left(\sum_{j=0}^{g-1} 2^j\right)\left(\sum_{i=0}^{h-1} 2^{ig}\right)}$$

so dass

$$\sum_{j=0}^{g-1} 2^j$$

$$\beta^{\sum_{j=0}^{g-1} 2^j} = \beta^{1+2+2^2+2^3.....2^{g-1}}$$

und wird als γ bezeichnet.

[0047] Dieser Term lässt sich mit dem Multiplizierer **48** gemäß Fig. 4 dadurch berechnen, dass man zunächst das Register **42** mit dem Wert X lädt. Dies wird um eine Zelle verschoben, um β (d.h. x^2) darzustellen, das Ergebnis wird in beide Register **42**, **44** geladen.

[0048] Das Register **44** wird dann verschoben, um β^2 zu erhalten, und die Registerinhalte der Register **42** und **44** werden multipliziert, um im Akkumulatorregister **46** den Wert β^{2+1} zu erhalten. Erreicht wird die Multiplikation mit einer Bewegung in Form einer m Bits umfassenden zyklischen Verschiebung jedes der Register **42**, **44** und

46.

[0049] Der akkumulierte Term β^{1+2} wird in die Register **44** und **42** transferiert, die β^2 enthalten, und dies wird um eine Stelle zum Erhalten von β^4 verschoben. Dann werden die Register **42**, **44** multipliziert, um β^{1+2+4} zu erhalten.

[0050] Diese Prozedur wird (g-2)-mal wiederholt, um γ zu erhalten. Wie im Folgenden beschrieben wird, lässt sich γ in ähnlicher Weise exponenzieren, um

$\gamma^{\sum_{i=0}^{h-1} 2^{ig}}$, d.h. x^{-1}

,d.h. x^{-1}, zu erhalten.

[0051] Dieser Term lässt sich ausdrücken in der Form

$\gamma^{1+2^g+2^{2g}+2^{3g}.....2^{(h-1)g}}$

.

[0052] Wie oben angemerkt, lässt sich γ dadurch auf **29** exponenzieren, dass man die Normalbasisdarstellung g-mal im Register **42** oder **44** verschiebt.

[0053] Damit werden die Register **42** und **44** jeweils mit dem Wert γ geladen, und das gmal verschobene Register **42** liefert den Wert γ^{2^g}. Die Register **42**, **44** werden multipliziert, um $\gamma \cdot \gamma^{2^g}$ oder γ^{1+2^g} zu erhalten im Akkumulatorregister **46**. Dieser Wert wird zum Register **44** transferiert, und das Register **42** wird g-mal verschoben, um $\gamma^{2^{2g}}$ zu erhalten.

[0054] Dann liefert die Multiplikation $\gamma^{1+2^g+2^{2g}}$. Das Wiederholen dieser Prozedur insgesamt (h-1)g-1-mal liefert die Inverse von X im Akkumulatorregister **46**.

[0055] Aus dem oben Gesagten ergibt sich, dass Quadrieren, Multiplizieren und Invertieren in effektiver Weise unter Verwendung des für endliche Körper ausgelegten Multiplizierers **48** erfolgen können.

Addition des Punkts p auf sich selbst (p + p) unter Verwendung von Unterroutinen

[0056] Um den Wert von dP zum Erzeugen des öffentlichen Schlüssels zu berechnen, berechnet die zum Empfänger **12** gehörige arithmetische Einheit **20** zunächst die Addition P + P. Wie eingangs erwähnt, besitzt bei einer nicht-supersingulären Kurve der neue Punkt Q Koordinaten (X_3, Y_3) mit

$$X_3 = X_1^2 \oplus \frac{b}{X_1^2}$$

$$Y_3 = X_1^2 \oplus (X_1 \oplus \frac{Y_1}{X_1}) X_3 \oplus X_3$$

[0057] Um X_3 zu berechnen, können gemäß Fig. 5 die folgenden Schritte implementiert werden.

[0058] Die m Bits, die X_1 darstellen, werden aus dem Basispunktregister **28** in das Register **42** geladen und eine Zelle nach rechts verschoben, um X_1^2 zu erhalten. Dieser Wert wird im RAM **36** gespeichert, und es wird in der oben beschriebenen Weise die Inverse von X_1^2 berechnet.

[0059] Der Wert von X_1^{-2} wird in das Register **44** geladen, und der Parameter b wird aus dem Parameterregister **30** geholt und in das Register **42** geladen. Das Produkt bX_1^{-2} wird im Akkumulatorregister **46** berechnet durch Drehen der Bit-Vektoren, und der Ergebniswert wird in dem Addierer **52** mit dem Wert X_1^2, der in dem RAM **36** gespeichert ist, eine Exklusiv-Oder-Verknüpfung unterzogen, um die Normalbasisdarstellung von X_3 zu erhalten. Das Ergebnis kann in dem RAM **36** gespeichert werden.

[0060] Eine ähnliche Prozedur kann dazu benutzt werden, Y_3 zu generieren, indem zunächst X_1 invertiert

wird, das Ergebnis mit Y multipliziert wird und dann in dem Addierer **52** mit X_1 einer Exklusiv-Oder-Verknüpfung unterzogen wird. Dies wird dann mit X_3 multipliziert, welches im RAM **36** gespeichert ist, und das Ergebnis wird exklusiv-oder-verknüpft mit dem Wert von X_3 und X_1^2, um Y_3 zu erhalten.

[0061] Der Ergebniswert von (X_3, Y_3) bedeutet die Summe P + P und ist ein neuer Punkt Q auf der Kurve. Dieser Wert kann dann auf P addiert werden, um einen neuen Punkt Q' zu bilden. Dieser Prozess kann (d-2)-mal wiederholt werden, um dP zu generieren.

[0062] Die Addition von P + Q erfordert die Berechnung von (X_3, Y_3) mit

$$x_3 = \left(\frac{y_1 \oplus y_2}{x_1 \oplus x_2}\right)^2 \oplus \frac{y_1 \oplus y_2}{x_1 \oplus x_2} \oplus x_1 \oplus x_2 \oplus a \quad .$$

und

$$y_3 = \left(\frac{y_1 \oplus y_2}{x_1 \oplus x_2}\right) (x_1 \oplus x_3) \oplus x_3 \oplus y_1$$

[0063] Dies wird (d-2)-mal mit einem neuen Wert für Q bei jeder Iteration wiederholt, um dP zu berechnen.

[0064] Während dies im Prinzip mit Hilfe der Arithmetikeinheit **20** geschehen kann, machen in der Praxis die großen Zahlen eine derartige Prozedur undurchführbar. Eine elegantere Vorgehensweise ist mit Hilfe der Binärdarstellung der ganzen Zahl d verfügbar.

Berechnung von dP aus 2P

[0065] Um das Addieren nicht-ähnlicher Punkte P und Q zu vermeiden, dient die Binärdarstellung von d in Verbindung mit einem Verdopplungsverfahren zum Reduzieren der Anzahl von Additionen und deren Komplexität.

[0066] Die ganze Zahl d lässt sich ausdrücken in der Form

$$d = \sum_{i=0} \lambda_i 2^i, \lambda_i \in (0,1) \quad \text{und} \quad dP = \sum_{i=0} \lambda_i (2^i P) \quad \text{d.h.}$$

$$\lambda_m 2^m P + \lambda_{m-1} 2^{m-1} P \ldots \lambda_3 2^3 P + \lambda_2 2^2 P + \lambda_1 2P + \lambda_0 P$$

[0067] Die Werte von λ sind die binäre Darstellung von d.

[0068] Nach der Berechnung von 2P kann der gewonnene Wert mit sich selbst addiert werden, wie oben in Verbindung mit **Fig. 5** beschrieben wurde, um 2^2P zu erhalten, welcher Wert wiederum auf sich selbst addiert werden kann, um 2^3P zu erhalten, etc. Dies wird solange wiederholt, bis 2^iP erhalten ist.

[0069] Bei jeder Iteration wird der Wert von 2^iP in dem RAM **36** gehalten für nachfolgende Additionen zwecks Gewinnung von dP.

[0070] Die arithmetische Einheit **20** führt eine weitere Menge von Additionen für nichtähnliche Punkte gleicher Terme durch, wobei λ den Wert 1 hat, so dass der Ergebniswert des Punkts (x_3, y_3) für dP erhalten wird.

[0071] Wenn beispielsweise k=5, so lässt sich dies berechnen in der Form 2^2P + p oder 2p + 2P + P oder Q + Q + P. Das Ergebnis lässt sich also durch drei Additionen erhalten. 2P = Q erfordert eine Addition, 2P + 2p = Q + Q = R bedeutet eine Addition, und R + P bedeutet eine Addition. Es sind höchstens t Verdopplungen und t anschließende Additionen erforderlich, abhängig davon, wie viele λ den Wert 1 haben.

Leistung der arithmetischen Einheit **20**

[0072] Für Berechnungen in einem Galois-Feld F_2^{155} hat sich gezeigt, dass das Berechnen der Inversen etwa 3800 Taktzyklen erfordert.

[0073] Das Verdoppeln eines Punkts, das ist die Addition eines Punkts auf sich selbst, nimmt größenordnungsmäßig 4500 Taktzyklen in Anspruch, und für eine praktische Implementierung eines privaten Schlüssels lässt sich die Berechnung des öffentlichen Schlüssels dP größenordnungsmäßig innerhalb von $1{,}5 \times 10^5$ Taktzyklen erhalten. Bei einer typischen Taktgeschwindigkeit von 40 MHz erfordert die Berechnung dP etwa 3×10^{-2} Sekunden. Dieser Durchsatz lässt sich steigern, wenn man den Seed-Schlüssel k mit einem Hamming-Gewicht, beispielsweise mit 20, begrenzt, um dadurch die Anzahl von Additionen nicht-ähnlicher Punkte einzugrenzen.

Berechnung eines öffentlichen Session-Schlüssels kP und des Chiffrierschlüssels kdP

[0074] Der öffentliche Session-Schlüssel kP lässt sich in ähnlicher Weise mit Hilfe der arithmetischen Einheit **20** des Senders **10** unter Verwendung des Basispunkts P aus dem Register **28** berechnen. Weil der öffentliche Schlüssel dP als ein Punkt (x_3, y_3) dargestellt wird, lässt sich in ähnlicher Weise auch der Chiffrierschlüssel kdP berechnen.

[0075] Jede dieser Operationen beansprucht ähnlich viel Zeit und kann vor dem Sendevorgang abgeschlossen werden.

[0076] Der Empfänger **12** muss in ähnlicher Weise dkP berechnen, wenn er den verschlüsselten Text C empfängt, was wiederum 3×10^{-2} Sekunden in Anspruch nimmt, also innerhalb der für eine praktische Implementierung einer Verschlüsselungseinheit erwarteten Zeit liegt.

[0077] Der öffentliche Schlüssel dP und der Session-Schlüssel kP werden jeweils dargestellt in Form einer 310 Bit langen Datenkette, und sie erfordern insoweit eine deutlich verringerte Bandbreite für die Übertragung. Gleichzeitig schaffen die Attribute von elliptischen Kurven eine sichere Verschlüsselungsstrategie mit praktischer Implementierung aufgrund der Effizienz der arithmetischen Einheit **20**.

Kurvenauswahl

a) Auswahl des Körpers F_q^m

[0078] Das obige Beispiel hat von einem Körper von 2^{155} Gebrauch gemacht sowie von einer nicht-supersingulären Kurve. Der Wert 155 wurde zum Teil deshalb gewählt, weil es eine optimale Normalbasis in F_2^{155} über F_2 gibt. Allerdings besteht eine Haupterwägung in der Sicherheit und der Effizienz des Verschlüsselungssystems. Der Wert 155 ist groß genug, um sicher zu sein, gleichzeitig aber auch klein genug, um einen effizienten Betrieb zu ermöglichen. Die Berücksichtigung üblicher Angriffe zum Knacken eines verschlüsselten Textes legt nahe, dass bei elliptischen Kurven über F_2^m ein Wert von m von etwa 130 ein sehr sicheres System ergibt. Die Verwendung von 1000 parallel arbeitenden Geräten ermöglicht das Auffinden eines Logarithmus in einer Zeit von etwa $1{,}5 \times 10^{11}$ Sekunden oder mindestens 1500 Jahren mit Hilfe des besten bekannten Verfahrens und des Körpers F_2^{155}. Andere Methoden führen zu noch längeren Laufzeiten.

b) Supersinguläre gegenüber nicht-supersingulären Kurven

[0079] Ein Vergleich von Angriffen auf mit Hilfe elliptischer Kurven verschlüsselte Daten legt nahe, dass nicht-supersinguläre Kurven robuster sind als supersinguläre Kurven. Für einen Körper F_g^h zeigt ein Angriff basierend auf dem Verfahren, wie es von Menezes, Okamoto und Vanstone in dem Artikel "Reducing elliptic curve logarithms to logarithms in finite field", veröffentlicht in Proceeding **22** Annual ACM Symposium Theory Computing 1991, Seiten 80–89 (The MOV attack) offenbart ist, dass bei kleinen Werten von k der Angriff subexponenziell wird. Zu den meisten supersingulären Kurven gehören kleine Werte von k. Im Allgemeinen jedoch haben nicht-supersinguläre Kurven große Werte von k, und bei $k>\log^2 q$ wird der MOV-Angriff weniger wirksam als mehr herkömmliche allgemeine Angriffe.

[0080] Die Verwendung einer supersingulären Kurve ist deshalb attraktiv, weil die Verdopplung eines Punkts (d.h. P = Q) keine Echtzeit-Inversion des zugrunde liegenden Körpers erfordert. Bei einer supersingulären Kurve lauten die Koordinaten von 2P sind

$$x_3 = \frac{x_1^4 \oplus b^2}{a^2} \quad \text{and} \quad y_3 = \left(\frac{x_1^2 \oplus b}{a}\right)(x_1 \oplus x_3) \oplus y_1 \oplus a.$$

[0081] Weil a eine Konstante ist, sind a^{-1} und a^{-2} für eine gegebene Kurve fest und können vorab berechnet

werden. Die Werte x_1^2 und x_1^4 lassen sich mit einer einfachen bzw. doppelten zyklischen Verschiebung im Multiplizierer **48** errechnen. Allerdings erfordert die anschließende Addition von nicht-ähnlichen Punkten zur Bildung des Werts von dP immer noch die Berechnung der Inversen in der Form

$$x_3 = \left(\frac{y_1 \oplus y_2}{x_1 \oplus x_2}\right)^2 \oplus x_1 \oplus x_2$$

und

$$y_3 = \left\{\left(\frac{y_1 \oplus y_2}{x_1 \oplus x_2}\right)(x_1 \oplus x_3) \oplus y_1 \oplus a\right.$$

[0082] Obwohl also supersinguläre Kurven zu effizienten Implementierungen führen, gibt es eine relativ kleine Menge von supersingulären Kurven, aus denen ausgewählt werden kann, insbesondere dann, wenn die Verschlüsselung robust sein soll. Bei einer supersingulären Kurve, bei der m ungerade ist, gibt es drei Klassen von Kurven, die weiter betrachtet werden können, nämlich

$y^2 + y = x^3$

$y^2 + y = x^3 + x$

$y^2 + y = x^3 + x + 1$

[0083] Allerdings zeigt eine Betrachtung dieser Kurven für den Fall m = 155, dass keine von ihnen die für Attacken notwendige Widerstandsfähigkeit besitzt.

[0084] Eine verbesserte Sicherheit für supersinguläre Kurven lässt sich dadurch erreichen, dass man quadratische Erweiterungen des zugrunde liegenden Körpers verwendet. Tatsächlich gibt es in F_q mit $q = 2^{310}$, d.h. einer quadratischen Erweiterung von F_2^{155} unter den supersingulären Kurven insgesamt vier, bei denen der MOV-Angriff die Berechnung diskreter log-Werte in F_2^{930} erfordert. Diese Kurven, liefern die erforderliche hohe Sicherheit und ermöglichen auch einen hohen Durchsatz. In ähnlicher Weise existieren in anderen Erweiterungen von Unterkörpern von F_2^{155} (z.B. F_2^{31}) weitere Kurven, die die erforderliche Robustheit besitzen. Allerdings erhöht ihre Verwendung die Anzahl der Ziffern, die einen Punkt definieren, mithin die Bandbreite bei ihrer Übertragung.

[0085] Im Gegensatz dazu beträgt die Anzahl nicht-supersingulärer Kurven von Fq mit $q = 2^{155}$:$2(2^{155} - 1)$. Durch Auswahl von q = 2, d.h. eines Körpers F_2^M, kann der Wert von a in der Darstellung der Kurve $y^2 + xy = x^3 + ax^2 + b$ so gewählt werden, dass er entweder 1 oder 0 lautet, ohne dass hierdurch ein Verlust von Allgemeingültigkeit entsteht. Diese umfangreiche Wahl von Kurven ermöglicht das Auffinden einer großen Anzahl von Kurven über diesem Körper, für die der Grad einer Kurve durch einen großen Primfaktor teilbar ist. Im Allgemeinen ist die Bestimmung des Grads einer beliebigen nicht-supersingulären Kurve über Fq nicht trivial, eine Möglichkeit ist ausgeführt in einem Papier mit dem Titel "Counting Points on Elliptic Curves" von Menezes, Vanstone und Zuccherato, Mathematics of Computation 1992.

[0086] Im Allgemeinen jedoch ist die Auswahl geeigneter Kurven im Stand der Technik bekannt. Als Beispiel wird verwiesen auf "Application of Finite Fields", Kapitel 7 und 8, von Menezes, Blake et al., Kluwer Academic Publishers (ISBN 0-7923-9282-5). Wegen der großen Anzahl derartiger Kurven, die den Anforderungen entsprechen, wird ungeachtet der zusätzlichen Berechnungen die Verwendung von nicht-supersingulären Kurven bevorzugt.

[0087] Eine alternative Vorgehensweise, welche die Anzahl von Inversionen bei Verwendung nicht-supersingulärer Kurven reduziert, besteht in der Verwendung von homogenen Koordinaten. Ein Punkt P wird definiert durch die Koordinaten (x, y, z) und Q durch den Punkt (x_2, y_2, z_2).

[0088] Der Punkt (0, 1, 0) bedeutet die Identität 0 in E.

[0089] Um die Additionsformen für die elliptische Kurve mit dieser Darstellung herzuleiten, nehmen wir die Punkte $P = (x_1, y_2, z_1)$ und $Q = (x_2, y_2, z_2)$, normieren jeweils auf $(x_1/z_1, y_1/z_1, 1)$, $(x_2/z_2, y_2/z_2, 1)$ und wenden die obigen Additionsformen an.
Wenn

$P = (x_1, y_1, z_1)$, $Q = (x_2, y_2, z_2)$, P, Q ≠ O, und P ≠ –Q, dann ist
$P + Q = (x_3, y_3, z_3)$, wobei, falls P ≠ Q, dann
$x_3 = AD$
$y_3 = CD + A^2(Bx_1 + Ay_1)$
$Z_3 = A^3Z_1Z_2$
wobei $A = x_2z_1 + x_1z_2$, $B = y_2z_1 + y_1z_2$, C = A + B und
$D = A^2(A + az_1z_2) + z_1z_2BC$.

[0090] Im Fall von P = Q gilt
$x_3 = AB$
$y_3 = x_1^4A + B(x_1^2 + y_1z_1 + A)$
$z_3 = A^3$ wobei $A = x_1z_1$ und $B = bz_1^4 + x_1^4$

[0091] Man sieht, dass die Berechnung von x_3, y_3 und z_3 keinerlei Inversion erfordert. Um allerdings die Koordinaten x_3^*, y_3^* in einer nicht-homogenen Darstellung herzuleiten, ist es notwendig, die Darstellung so zu normieren, dass

$$x_3^* = \frac{x_3}{z_3} \qquad y_3^* = \frac{y_3}{z_3}$$

[0092] Dieser Vorgang erfordert eine Inversion, die die oben angesprochene Prozedur verwendet. Allerdings ist lediglich eine Inversion zum Berechnen von dP erforderlich.

[0093] Durch die Verwendung homogener Koordinaten ist es immer noch möglich, dP unter Verwendung der Version des Verdopplungs- und Additionsverfahrens zu berechnen, welches oben beschrieben wurde. Der Berechnungsvorgang von P + Q, P ≠ Q, erfordert 13 Körper-Multiplikationen, und 2P erfordert 7 Multiplikationen.

Alternativer Schlüsseltransfer

[0094] In dem obigen Beispiel werden die Koordinaten der Schlüssel kP transferiert als zwei 155 Bit lange Körperelemente für F_2^{155}. Um die Bandbreite noch weiter zu reduzieren, ist es möglich, nur eine der Koordinaten zu senden und die anderen Koordinaten im Empfänger zu berechnen. Eine Kennung, beispielsweise ein einzelnes Bit des korrekten Werts der anderen Koordinate, kann ebenfalls übertragen werden. Dies ermöglicht, die zweite Koordinate am Empfänger zu berechnen und die korrekte Koordinate anhand der Kennung zu identifizieren.

[0095] Bezugnehmend auf **Fig. 1** holt also der Sender **10** zunächst den öffentlichen Schlüssel dP des Empfängers **12**, ferner eine Bit-Kette für die Koordinate x_0 und ein einzelnes Bit der Koordinate y_0.

[0096] Der Sender **10** besitzt die Parameter der Kurve im Register **30** und kann daher die Koordinate x_0 und die Kurven-Parameter dazu benutzen, mögliche Werte der anderen Koordinate y_0 aus der arithmetischen Einheit **20** zu erhalten.

[0097] Für eine Kurve in der Form $y^2 + xy = x^3 + ax^2 + b$ und eine Koordinate x_0 sind dann die möglichen Werte y_1, y_2 für y_0 die Wurzeln der quadratischen Gleichung $y^2 + x_0Y = x_0^3 + ax_0^2 + b$.

[0098] Durch Auflösen nach y in der arithmetischen Einheit **20** erhält man zwei mögliche Wurzeln, und ein Vergleich mit dem gesendeten Informations-Bit gibt an, welcher der Werte der passende Wert für y ist.

[0099] Die beiden möglichen Werte der zweiten Koordinate (y_0) unterscheiden sich um x_0, d.h. $y_1 = y_2 + x_0$.

[0100] Da die beiden Werte von y_0 um x_0 voneinander abweichen, unterscheiden sich y, und y_2 stets, wobei "1" in der Darstellung von x_0 auftritt. Folglich wird das zusätzlich gesendete Bit aus einer jener Positionen ausgewählt, und die Untersuchung des entsprechenden Bits der Werte von y_0 gibt an, welche der beiden Wurzeln der passende Wert ist.

[0101] Auf diese Weise kann der Empfänger **10** die Koordinaten des öffentlichen Schlüssels dP auch dann generieren, wenn nur 156 Bits aufgefunden werden.

[0102] Ähnliche Wirkungsweisen lassen sich beim Senden des Session-Schlüssels kP zu dem Empfänger **12** realisieren, da der Sender **12** nur eine Koordinate übertragen muss, nämlich x_0 und das ausgewählte Ken-

nungsbit für y_0. Der Empfänger **12** kann dann die möglichen Werte von y_0 rekonstruieren und den passenden Wert wählen.

[0103] Im Körper F_2^m ist es nicht möglich, mit Hilfe der quadratischen Formel bei 2a = 0 eine Lösung für y zu erhalten. Folglich werden andere Methoden benötigt, wobei die arithmetische Einheit **20** speziell ausgebildet ist, um dies effizient auszuführen.

[0104] Im Allgemeinen ist x_0 von null verschieden, und wenn $y = x_0 z$, so gilt $x_0^2 z^2 + x_0^2 z = x_0^3 + a x_0^2 + b$

[0105] Dies lässt sich umschreiben in der Form

$$z^2 + z = x_0 + a + \frac{b}{x_0^2} = c.$$

[0106] Das heißt: $z^2 + z = c$.

[0107] Wenn m ungerade ist, so gilt entweder

$$z = c + c^4 + c^{16} .... + ... + c^{2^{m-1}}$$

$$= c^{2^0} + c^{2^2} + c^{2^4} + ... + c^{2^{m-1}}$$

oder

$$z = 1 + c^{2^0} + ... + c^{2^{m-1}}$$

um zwei mögliche Werte für y_0 zu schaffen.

[0108] Eine ähnliche Lösung existiert für den Fall, dass m gerade ist, so dass ebenfalls die Terme in der Form c^{20} benutzt werden.

[0109] Dies eignet sich besonders zur Verwendung bei einer Normalbasisdarstellung in F_2^m

[0110] Wie oben angemerkt, lässt sich das Erheben eines Köperelements in F_2^m in die Potenz von g erreichen durch eine g-fache zyklische Verschiebung, bei der das Körperelement als eine Normalbasis dargestellt wird.

[0111] Folglich lässt sich jeder Wert von z berechnen, indem man Verschiebungen und Additionen durchführt, um die Wert für y_0 zu erhalten. Der korrekte Wert bestimmt sich durch das zusätzlich übertragene Bit.

[0112] Die Verwendung einer Normalbasisdarstellung in F_2^m vereinfacht mithin das Protokoll zum Wiedergewinnen der Koordinate y_0.

[0113] Wenn $P = (x_0 y_0)$ ein Punkt auf der elliptischen Kurve $E : y^2 + xy = x^3 + ax^2 + b$ ist, definiert über einem Körper F_2^m, dann ist $\hat{y}_0$ definiert zu 0, falls $x_0 = 0$, und bei $x_0 \neq 0$ ist $\hat{y}_0$ definiert durch das niedrigstwertige Bit in dem Körperelement $y_0 \cdot x_0^1$.

[0114] Die x-Koordinate x_0 von P und das Bit $\hat{y}_0$ werden zwischen dem Sender **10** und dem Empfänger **12** übertragen. Die y-Koordinate y_0 lässt sich dann folgendermaßen gewinnen.

1. Wenn $x_0 = 0$, so wird y_0 durch zyklisches Verschieben der Vektordarstellung des Körperelements b, die in dem Parameterregister 30 um eine Position nach links gespeichert ist, erhalten. Das heißt, wenn $b = b_{m-1} b_{m-2} .... b_1 b_0$ dann $y_0 = b_{m-2} .... b_1 b_0 b_{m-1}$
2. Wenn $x_0 \neq 0$, dann geschieht Folgendes:

2.1 Berechne das Körperelement $c = x_0 + a + b x_0^{-2}$ in F_2^m

2.2 Sei die Vektordarstellung von c $C = C_{m-1} C_{m-2} .... C_1 C_0$.

2.3 Konstruiere ein Körperelement $z = Z_{m-1} Z_{m-2} .... Z_1 Z_0$, indem gesetzt wird: $Z_0 = y_0$, $Z_1 = C_0 \oplus Z_0$, $Z_2 = C_1 \oplus Z_1$, $Z_{m-2} = C_{m-3} \oplus Z_{m-3}$,

2.4 Schließlich berechne man $y_0 = x_0 \cdot z$.

[0115] Man sieht, dass die Berechnung x_0^{-1} in einfacher Weise in der arithmetischen Einheit **20** in oben beschriebener Weise vorgenommen werden kann, und dass die Berechnung von y_0 mit Hilfe des Multiplizierers **48** vorgenommen werden kann.

[0116] In den obigen Beispielen wurde die Identifizierung des richtigen Werts von y_0 erreicht durch Übertragen eines einzelnen Bits und durch einen Vergleich des Werts der gewonnenen Wurzel. Allerdings können auch andere Indikatoren zum Identifizieren des passenden Werts verwendet werden, wobei der Vorgang nicht beschränkt ist auf die Verschlüsselung mit Hilfe elliptischer Kurven auf dem Körper $GF(2^m)$. Wenn beispielsweise der Körper ausgewählt ist zu Zp p = 3 (mod 4), so könnte das Legendre-Symbol in Verbindung mit dem passenden Wert gesendet werden, um den richtigen Wert zu kennzeichnen. Alternativ könnte die Menge der Elemente in Zp aufgeteilt werden in ein Paar Untermengen mit der Eigenschaft, dass, wenn y in der einen Untermenge liegt, -y in der anderen liegt, vorausgesetzt, dass y≠0. Dann lässt sich den einzelnen Untermengen ein beliebiger Wert zuordnen und mit der Koordinate x_0 übertragen, um anzugeben, in welcher Untermenge sich der passende Wert für y_0 befindet. Folglich lässt sich der passende Wert für y_0 bestimmen. In geeigneter Weise ist es möglich, eine passende Darstellung zu verwenden, bei der die Untermengen angeordnet sind als Intervalle, um so die Kennzeichnung des passenden Werts von y_0 zu erleichtern.

[0117] Diese Methoden eignen sich besonders gut für die Verschlüsselung unter Verwendung elliptischer Kurven, lassen sich aber auch einsetzen bei beliebigen algebraischen Kurven und finden Anwendung auf anderen Gebieten, so z.B. bei einer Fehlerkorrekturkodierung, bei der Koordinaten von Punkten auf Kurven zu transferieren sind.

[0118] Man erkennt daher, dass durch Verwenden einer elliptischen Kurve, die in dem endlichen Körper GF_2^m liegt, und durch Verwenden einer Normalbasisdarstellung die zum Verschlüsseln mit elliptischen Kurven notwendigen Berechnungen effizient ausgeführt werden können. Derartige Berechnungen lassen sich entweder in Software oder in Hardware implementieren, und die Strukturierung der Berechnungen macht Gebrauch von einem Multiplizierer für endliche Körper, was insbesondere bei Implementierung als Hardware effizient ist.

Patentansprüche

1. Verfahren zum Transferieren von Koordinaten eines Punkts auf einer nicht supersingulären elliptischen Kurve von einem ersten Korrespondenten (**10**) zu einem zweiten Korrespondenten (**12**), welcher mit dem ersten Korrespondenten durch eine Datenübertragungsverbindung (**12**) verbunden ist und die Parameter der Kurve enthält, wobei die Koordinaten eine erste Koordinate beinhalten, die zwei mögliche Punkte auf der Kurve bestimmen, die jeweils beide eine zugehörige zweite Koordinate besitzen, umfassen folgende Schritte:
a) der erste Korrespondent wird entsprechend der ersten Koordinate des Punkts zu dem zweiten Korrespondenten vorgerückt;
b) der erste Korrespondent identifiziert für den zweiten Korrespondenten eine Untermenge, die eine der zweiten Koordinaten entsprechend dem genannten Punkt beinhaltet und die andere der zweiten Koordinaten ausschließt;
c) der zweite Korrespondent berechnet die zweite Koordinate aus der ersten Koordinate und der Kurve;
d) der zweite Korrespondent bestimmt, welche der zweiten Koordinaten in der identifizierten Untermenge enthalten ist, um dadurch den Wert der einen der zweiten Koordinaten entsprechend dem genannten Punkt zu bestimmen.

2. Verfahren nach Anspruch 1, bei dem für jedes in einer der Untermengen enthaltenen, von null verschiedenen Element, eine Negation des Elements nicht in der Untermenge enthalten ist.

3. Verfahren nach Anspruch 1 oder 2, bei dem die Untermenge ein Intervall ist.

4. Verfahren nach einem der Ansprüche 1 und 3, bei dem der Schritt (b) die Übertragung eines der identifizierten Untermenge zugewiesenen Werts beinhaltet.

5. Verfahren nach einem vorhergehenden Anspruch, bei dem die algebraische Kurve über dem Feld Zp definiert ist.

6. Verfahren zum Transferieren von Koordinaten eines Punkts auf einer nichtsupersingulären elliptischen Kurve von einem ersten Korrespondenten (**10**) zu einem zweiten Korrespondenten (**12**), welcher mit dem ersten Korrespondenten durch eine Datenübertragungsverbindung (**14**) verbunden ist und die Parameter der Kurve enthält, wobei die Koordinaten eine erste Koordinate beinhalten, die zwei mögliche Punkte auf der Kurve bestimmen, die jeweils beide eine zugehörige zweite Koordinate besitzen, umfassen folgende Schritte:
a) der erste Korrespondent wird entsprechend der ersten Koordinate des Punkts zu dem zweiten Korrespondenten vorgerückt; und
b) der erste Korrespondent identifiziert für den zweiten Korrespondenten eine Untermenge, die eine der zwei-

ten Koordinaten entsprechend dem genannten Punkt enthält, schließt die andere der zweiten Koordinaten aus; wobei der zweite Korrespondent die zweiten Koordinaten berechnen kann aus der ersten Koordinate und der Kurve, und bestimmen kann, welche der zweiten Koordinaten in der identifizierten Untermenge enthalten ist, um dadurch den Wert der einen der zweiten Koordinaten entsprechend dem genannten Punkt zu bestimmen.

7. Verfahren nach Anspruch 6, bei dem für jedes in der Untermenge enthaltene, von null verschiedene Element eine Negation des Elements nicht in der Untermenge enthalten ist.

8. Verfahren nach Anspruch 6 oder 7, bei dem die Untermenge ein Intervall ist.

9. Verfahren nach einem der Ansprüche 6 bis 8, bei dem der Schritt (b) das Übertragen eines der identifizierten Untermenge zugeordneten Werts beinhaltet.

10. Verfahren nach Anspruch 6 bis 9, bei dem die algebraische Kurve über dem Feld Zp definiert ist.

11. Verfahren zum Transferieren von Koordinaten eines Punkts auf einer nichtsupersingulären elliptischen Kurve von einem ersten Korrespondenten (**10**) zu einem zweiten Korrespondenten (**12**), welcher mit dem ersten Korrespondenten durch eine Datenübertragungsverbindung (**14**) verbunden ist und die Parameter der Kurve enthält, wobei die Koordinaten eine erste Koordinate beinhalten, die zwei mögliche Punkte auf der Kurve bestimmen, die jeweils beide eine zugehörige zweite Koordinate besitzen, umfassen folgende Schritte:
a) der zweite Korrespondent empfängt von dem ersten Korrespondenten die erste Koordinate des genannten Punkts;
b) der zweite Korrespondent empfängt von dem ersten Korrespondenten eine Identifizierung einer Untermenge einer von den zweiten Koordinaten entsprechend im genannten Punkt und schließt die andere der zweiten Koordinaten aus;
c) der zweite Korrespondent berechnet die zweiten Koordinaten aus der ersten Koordinate und der Kurve;
d) der zweite Korrespondent bestimmt, welche der zweiten Koordinaten in der identifizierten Untermenge enthalten ist, um dadurch den Wert der einen der zweiten Koordinaten entsprechend dem genannten Punkt zu bestimmen.

12. Verfahren nach Anspruch 11, bei dem für jedes in der Untermenge enthaltene, von null verschiedene Element eine Negation des Elements nicht in der Untermenge enthalten ist.

13. Verfahren nach Anspruch 11 oder 12, bei dem die Untermenge ein Intervall ist.

14. Verfahren nach einem der Ansprüche 11 bis 13, bei dem der Schritt (b) das Empfangen eines der identifizierten Untermenge zugeordneten Werts beinhaltet.

15. Verfahren nach einem der Ansprüche 11 bis 14, bei dem die algebraische Kurve über dem Feld Zp definiert ist.

16. System (**10**, **12**, **14**), umfassend einen Sender (**10**) und einen Empfänger (**12**) sowie eine Übertragungsverbindung (**14**), die Sender und Empfänger zum Übertragen von Koordinaten eines Punkts auf einer nicht-supersingulären elliptischen Kurve von einem ersten Korrespondenten (**10**) in dem Sender zu einem zweiten Korrespondenten (**12**) in dem Empfänger und mit den Parametern der Kurve verbindet, wobei die Koordinaten eine erste Koordinate beinhalten die zwei mögliche Punkte auf der Kurve bestimmen, wobei jeder der möglichen Punkte eine zugehörige zweite Koordinate besitzt, und der Sender aufweist:
a) eine Liefereinrichtung zum Liefern der ersten Koordinate des Punkts zu dem zweiten Korrespondenten;
b) eine Identifiziereinrichtung zum Identifizieren einer die eine der zweiten Koordinaten entsprechend dem genannten Punkt enthaltende Untermenge für den zweiten Korrespondenten unter Ausschluss der anderen der zweiten Koordinaten;
wobei der Empfänger aufweist:
c) eine Berechnungseinrichtung zum Berechnen der zweiten Koordinate aus der ersten Koordinate und der Kurve; und
d) eine Bestimmungseinrichtung zum Bestimmen, welche der zweiten Koordinaten in der identifizierten Untermenge enthalten ist, um dadurch den Wert der einen der zweiten Koordinaten entsprechend dem genannten Punkt zu bestimmen.

17. System nach Anspruch 16, bei dem der Sender eine Einrichtung aufweist zum Senden eines der identifizierten Untermenge zugeordneten Werts.

18. Sender zur Verwendung bei dem Transferieren von Koordinaten eines Punkts auf einer nicht-supersingulären elliptischen Kurve von einem ersten Korrespondenten (**10**) zu einem zweiten Korrespondenten (**12**), der mit dem ersten Korrespondenten über eine Datenübertragungsverbindung (**14**) verbunden ist und die Parameter der Kurve enthält, wobei die Koordinaten eine erste Koordinate enthalten, welche zwei mögliche Punkte auf der Kurve bestimmen, und jeder der möglichen Punkte eine zugehörige zweite Koordinate besitzt, wobei der Sender aufweist:
a) eine Liefereinrichtung zum Liefern der ersten Koordinate des genannten Punkts zu dem zweiten Korrespondenten;
b) eine Identifiziereinrichtung zum Identifizieren einer eine der zweiten Koordinaten entsprechend dem genannten Punkt enthaltende Untermenge für den zweiten Korrespondenten unter Ausschluss der anderen der zweiten Koordinaten;
wobei der zweite Korrespondent die zweiten Koordinaten berechnen kann aus der ersten Koordinate und der Kurve, und bestimmen kann, welche der zweiten Koordinaten in der identifizierten Untermenge enthalten ist, um dadurch den Wert der einen der zweiten Koordinaten entsprechend dem genannten Punkt zu bestimmen.

19. Empfänger zur Verwendung bei dem Transferieren von Koordinaten eines Punkts auf einer nicht-supersingulären elliptischen Kurve von einem ersten Korrespondenten (**10**) zu einem zweiten Korrespondenten (**12**) über eine Datenübertragungsverbindung (**14**) verbunden ist und die Parameter der Kurve enthält,
wobei die Koordinaten eine erste Koordinate enthalten, welche zwei mögliche Punkte auf der Kurve bestimmen, und jeder der möglichen Punkte eine zugehörige zweite Koordinate besitzt, wobei der Empfänger aufweist:
a) eine Einrichtung zum Empfangen der ersten Koordinate des genannten Punkts von dem ersten Korrespondenten;
b) eine Einrichtung zum Empfangen einer Identifizierung einer Untermenge einer der zweiten Koordinaten entsprechend dem genannten Punkt unter Ausschluss der anderen der zweiten Koordinaten von dem ersten Korrespondenten;
c) eine Berechnungseinrichtung zum Berechnen der zweiten Koordinaten aus der ersten Koordinate und der Kurve;
d) eine Bestimmungseinrichtung zum Bestimmen, welche der zweiten Koordinaten in der identifizierten Untermenge enthalten ist, um dadurch den Wert der einen der zweiten Koordinaten entsprechend dem genannten Punkt zu bestimmen.

20. Computerprogramm mit durch einen Prozessor implementierbaren Schritten, ausgebildet zum Durchführen eines Verfahrens nach einem der Ansprüche 1 bis 15.

Es folgen 5 Blatt Zeichnungen

Anhängende Zeichnungen

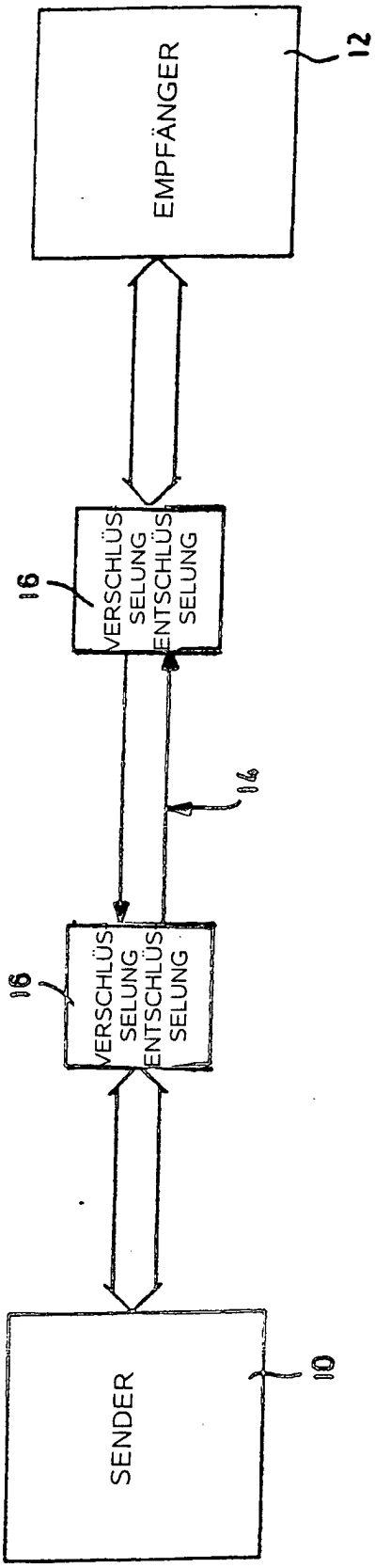

FIG. 1

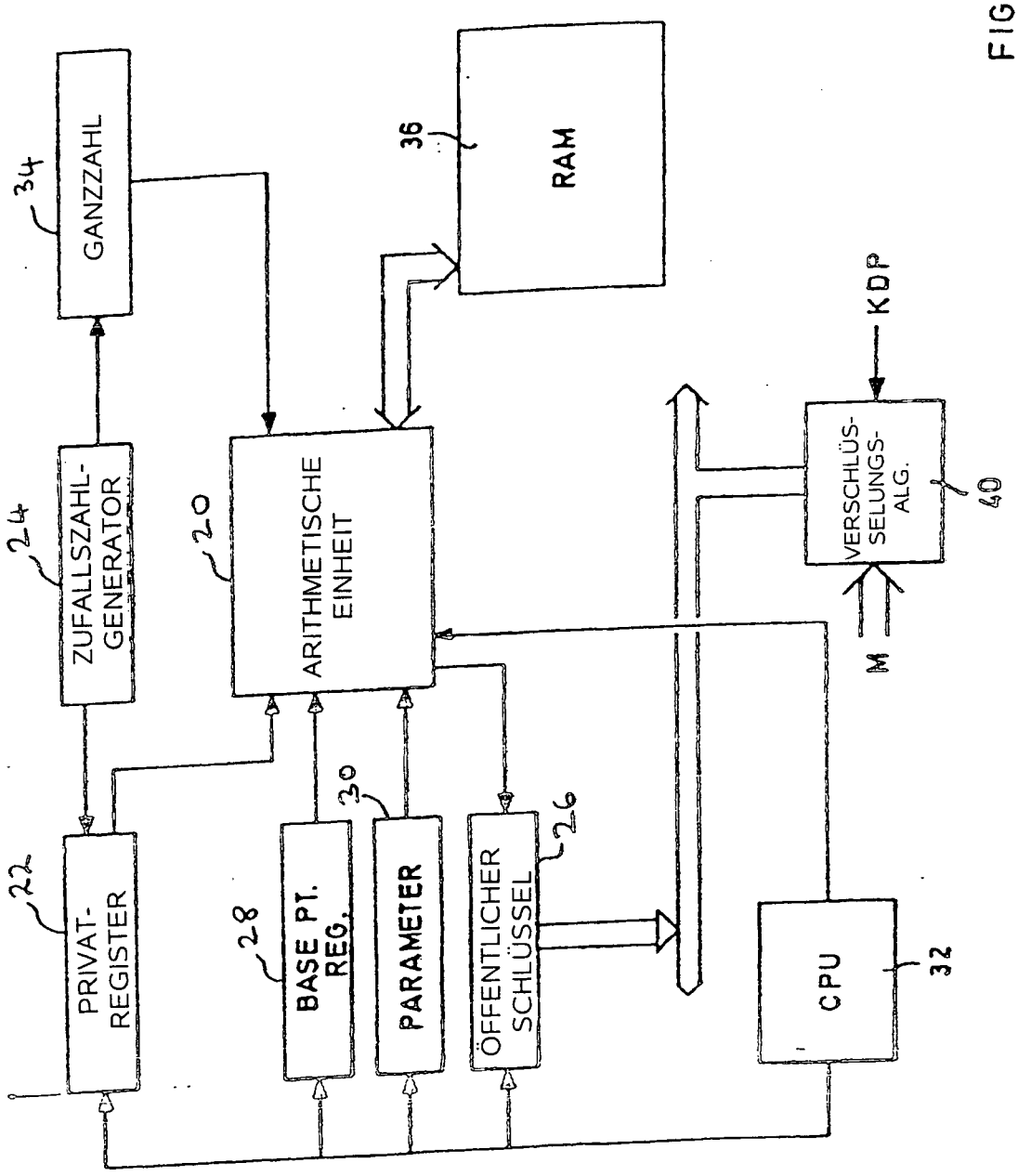
GANZZAHL
34
RAM
36
KDP
ZUFALLSZAHL-GENERATOR
24
ARITHMETISCHE EINHEIT
20
VERSCHLÜSSELUNGS-ALG.
40
M
PRIVAT-REGISTER
22
BASE PT. REG.
28
PARAMETER
30
ÖFFENTLICHER SCHLÜSSEL
26
CPU
32

FIG. 2

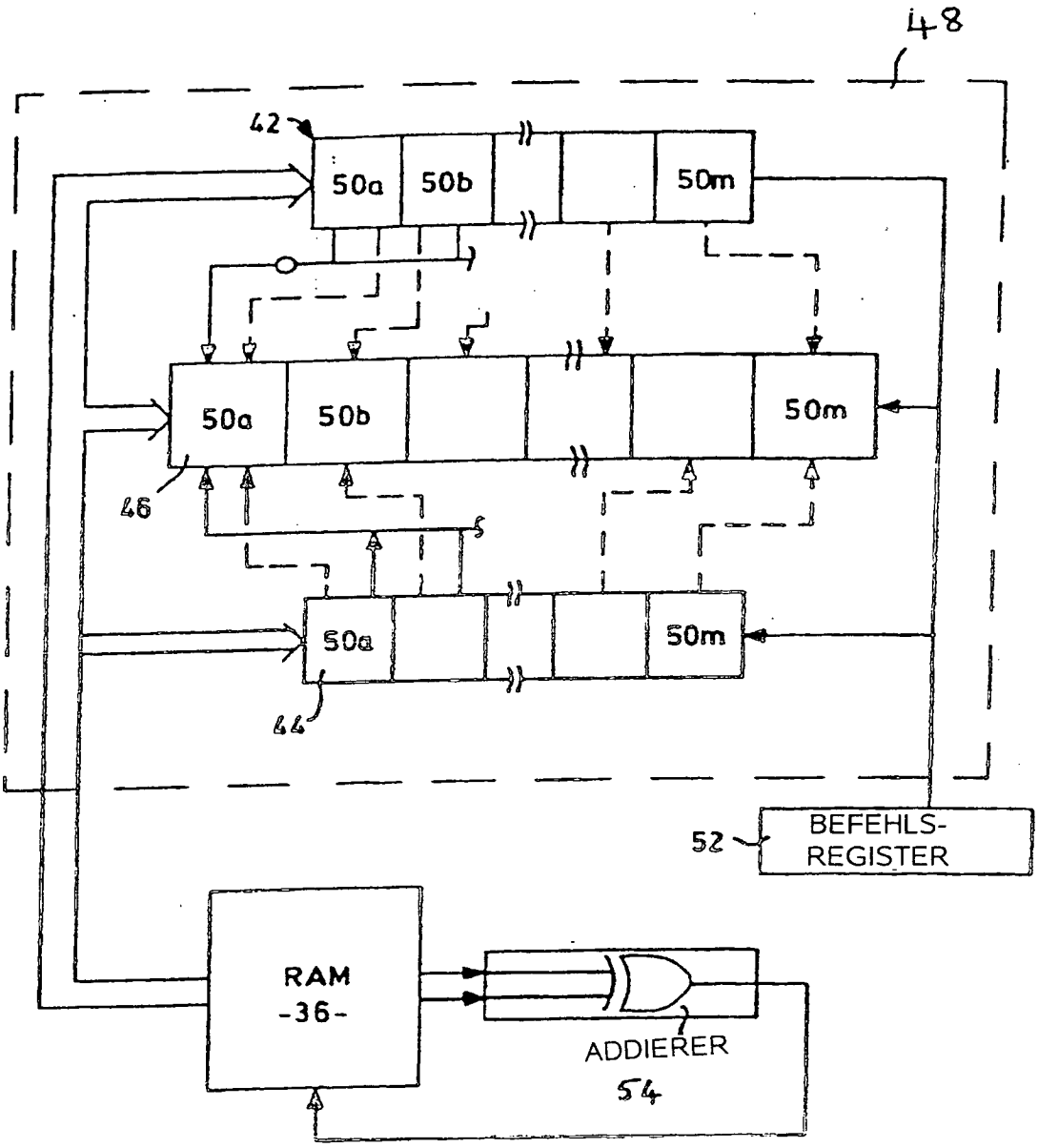
48
42
50a
50b
50m
50a
50b
50m
46
50a
50m
44
52
BEFEHLS-
REGISTER
RAM
-36-
ADDIERER
54

FIG. 3

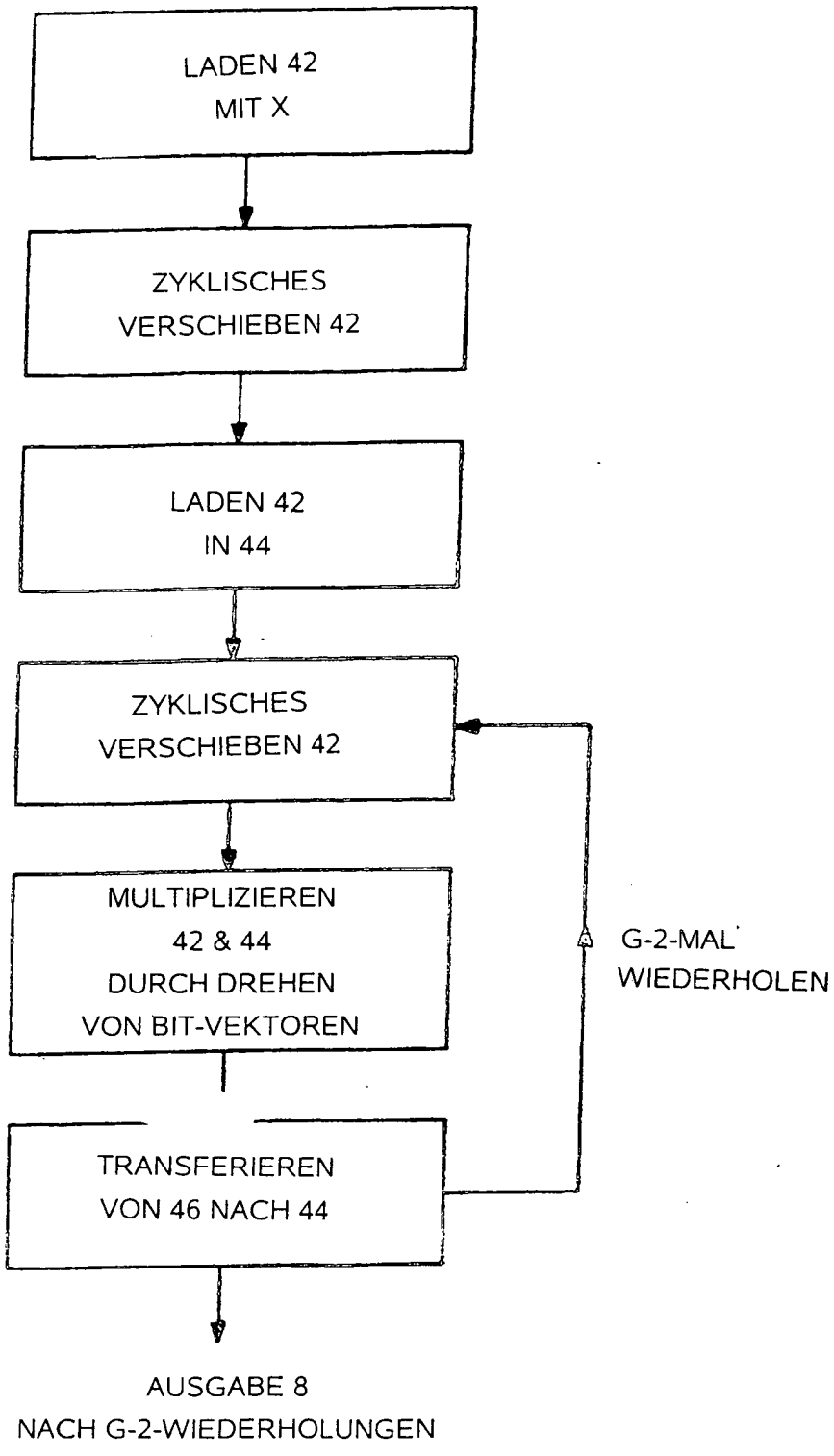
LADEN 42
MIT X
ZYKLISCHES
VERSCHIEBEN 42
LADEN 42
IN 44
ZYKLISCHES
VERSCHIEBEN 42
MULTIPLIZIEREN
42 & 44
DURCH DREHEN
VON BIT-VEKTOREN
TRANSFERIEREN
VON 46 NACH 44
G-2-MAL
WIEDERHOLEN
AUSGABE 8
NACH G-2-WIEDERHOLUNGEN

FIG. 4

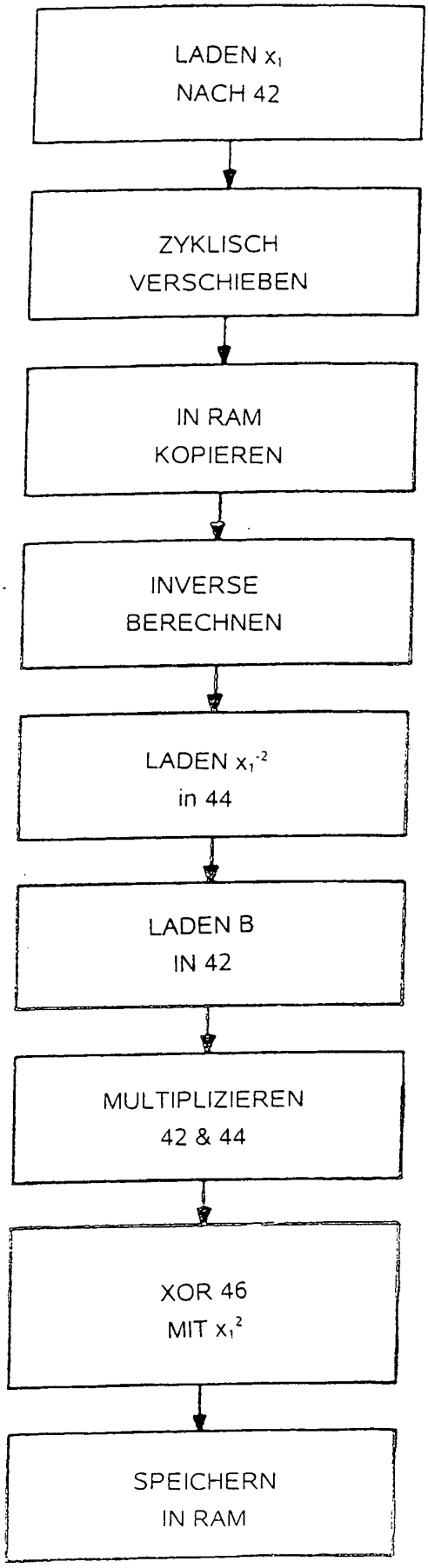
LADEN x_1
NACH 42
ZYKLISCH
VERSCHIEBEN
IN RAM
KOPIEREN
INVERSE
BERECHNEN
LADEN x_1^{-2}
in 44
LADEN B
IN 42
MULTIPLIZIEREN
42 & 44
XOR 46
MIT x_1^2
SPEICHERN
IN RAM

FIG. 5