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

(10) **DE 601 27 747 T2** 2008.01.24

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 136 903 B1**
(21) Deutsches Aktenzeichen: **601 27 747.3**
(96) Europäisches Aktenzeichen: **01 480 017.1**
(96) Europäischer Anmeldetag: **27.02.2001**
(97) Erstveröffentlichung durch das EPA: **26.09.2001**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **11.04.2007**
(47) Veröffentlichungstag im Patentblatt: **24.01.2008**

(51) Int Cl.[8]: ***G06F 21/00*** (2006.01)

(30) Unionspriorität:
00480030 20.03.2000 EP

(84) Benannte Vertragsstaaten:
AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI, LU, MC, NL, PT, SE, TR

(73) Patentinhaber:
International Business Machines Corp., Armonk, N.Y., US

(74) Vertreter:
Duscher, R., Dipl.-Phys. Dr.rer.nat., Pat.-Ass., 70176 Stuttgart

(72) Erfinder:
Incertis Carro, Fernando, 46182 Valencia, ES;
Matyas, Stephen, Manassa, VA 20110, US

(54) Bezeichnung: **Verfahren und System zur reversiblen Markierung eines Textdokuments mit einem Muster der zusätzlichen Leerzeichen für Beglaubigung**

Beschreibung

Gebiet der Erfindung

[0001] Die vorliegende Erfindung bezieht sich auf das Gebiet des Nachweises der Identität von Dokumenten. Genauer befasst sie sich mit dem Nachweis der Identität von elektronisch kopierten Textdokumenten.

Hintergrund der Erfindung

[0002] Im derzeitigen Umfeld der Computernetzwerke, das von einem exponentiellen Wachstum des Umlaufes von elektronisch kopierten oder elektronischen Textdokumenten wie etwa eMail über ungesicherte Medien, z.B. das Internet, charakterisiert ist, ist der Nachweis der Identität (authentication) ein Hauptproblem. Es sollte für den Empfänger eines Textdokumentes möglich sein, sich dessen Ursprungs zu versichern, so dass niemand sich als jemand anderes ausgeben kann. Auch sollte es möglich sein, zu überprüfen, ob es unterwegs nicht zufällig oder böswillig geändert wurde. Zu diesem Zweck wurden Verfahren entwickelt, um den Nachweis der Identität auszuführen.

[0003] Die Standardlösung, die für elektronische Textdokumente gut geeignet ist, besteht darin, elektronisch kopierten Textdokumenten einen MAC (Message Authentification Code) oder Nachrichten-Identitätsnachweiscode hinzuzufügen. Ein MAC ist eine Zusammenfassung, die von einer nicht umkehrbaren Hash-Funktion auf den Text angewendet berechnet wird und die auch von einem Schlüssel, z.B. einem nur dem Absender und Empfänger bekannten Geheimschlüssel, abhängig gemacht werden kann, damit der Empfänger erstens prüfen kann, ob das Empfangene in der Tat vom Teilhaber des Geheimschlüssels versendet wurde, und zweitens, dass das Dokument nicht geändert worden ist. Z.B. erzeugt der Secure Hash Algorithmus oder SHA, spezifiziert vom National Institute of Standards and Technologies, NIST, FIPS PUB 180-1, „Secure Hash Standard", US Dpt. Of Commerce, May 93, einen 160-Bit-Hash-Code. Er kann mit einem Schlüssel verknüpft werden, z.B. durch die Anwendung eines Mechanismus, der als HMAC oder Keyed-Hashing zum Nachrichten-Identitätsnachweis bezeichnet wird und Gegenstand der RFC (Request for Comment) bei der IETF (Internet Engineering Task Force) unter der Nummer 2104 ist. HMAC ist so konzipiert, dass er mit jeder iterativen kryptographischen Hashfunktion einschließlich SHA verwendet werden kann. Somit kann ein MAC an die elektronische Kopie eines Textdokuments angefügt werden, womit das Gesamte vom Empfänger geprüft werden kann. So geht das Verfahren davon aus, dass die Prüfinformationen der Datei hinzugefügt werden, mit dem Nachteil allerdings, dass Text und Prüfinformationen getrennt werden. Daher kann diese Letztere auf einfache Weise absichtlich isoliert und entfernt werden, als Versuch des Betruges oder rein zufällig, weil dazwischen liegende Teile der Ausrüstung, die die Weiterleitung der elektronischen Dokumente übernehmen, nicht die Handhabung dieser zusätzlichen Informationen vorsehen.

[0004] Dann sollten die Prüfinformationen vielmehr transparent beispielsweise im Hauptteil des Textdokuments selbst so verschlüsselt sein, dass die Lesbarkeit und dergleichen nicht beeinträchtigt werden, damit es bei den verschiedenen Manipulationen, denen es auf dem Weg zu seinem Ziel ausgesetzt ist, intakt bleibt und der Empfänger am Ende die Identität des Dokuments nachweisen kann.

[0005] Ein weiterer Ansatz für den Nachweis der Identität, der hauptsächlich für elektronisch kopierte Bilder (und somit auch für das Bild eines Textdokumentes) verwendbar ist, besteht darin, Daten in ihrer entsprechenden digitalen Darstellung zu verstecken, womit die obige Anforderung erfüllt ist, die Prüfinformationen bevorzugt im Dokument selbst unterzubringen. Das Verstecken von Daten in Bildern hat beträchtliche Aufmerksamkeit erfahren, hauptsächlich aufgrund der Urheberrechte, die mit digitalem Multimediamaterial verbunden sind, das leicht kopiert und überall durch das Internet und Netzwerke allgemein verbreitet werden kann. Eine gute Übersicht über Techniken des Versteckens von Daten findet sich in „Techniques for data hiding" von W. Bender et al., publiziert im IBM Systems Journal, Band 35, Nr. 3 und 4, 1996. Als eine Veranschaulichung für die Vorgehensweise, Daten zu verstecken, dient als die häufigste Form der Verschlüsselung mit hoher Bit-Rate, die hier weiter oben beschrieben wurde, das Ersetzen des niedrigstwertigen Luminanzbits der Bilddaten mit den eingebetteten Daten. Diese Technik, die tatsächlich die Anforderung erfüllt, nicht wahrnehmbar zu sein (das wiederhergestellte Bild ist weit davon entfernt, erkennbar geändert worden zu sein), kann ähnlich einem Nachweis der Identität, z.B. mittels Wasserzeichen, das einem Bild ein unveränderliches Kennzeichen verleiht, oder einer Prüfung auf Manipulation zur Erkennung von Bildveränderungen, durch das Einbetten eines MAC in die elektronische Bildkopie verschiedenen Absichten dienen.

[0006] Allerdings wäre die notwendige Betrachtung von Text als Bild eine sehr kostspielige und ungeeignete Lösung in Hinblick auf Speicher und der zur Übertragung erforderlichen Bandbreite. Obwohl wie hier weiter oben behauptet elektronisch kopierter Text in vielerlei Hinsicht aufgrund des Fehlens von redundanten Informationen in einer Textdatei im Vergleich zu einer Bilddatei der schwierigste Ort ist, Daten zu verstecken, ist die Manipulation von weißen Lücken, d.h. von Leerzeichen und speziell von absichtlich vom Verfasser eines Textdokumentes eingefügten Leerzeichen zwischen Wörtern, in krassem

Gegensatz zu dem, was zur Lesbarmachung eines Textdokumentes zwingend erforderlich ist (d.h. ein Leerzeichen zwischen je zwei Wörtern), die einfachste Art und Weise, einen dem Nachweis der Identität ausgesetzten Text ohne Anfügen eines separaten MAC zu markieren, da die zur Prüfung notwendige Informationen dann im Text selbst in Form von zusätzlichen Leerzeichen zwischen Wörtern gewissermaßen so versteckt eingebettet ist, dass die Wahrscheinlichkeit gering ist, dass ein gewöhnlicher Leser davon Kenntnis nimmt. Darüber hinaus sollte der Endempfänger des Dokumentes idealerweise (selbst wenn der Text lesbar ist) in der Lage sein, das Original-Textdokument genau so zurückzuformatieren, wie es erstellt wurde. Das Hinzufügen von zusätzlichen Leerstellen sollte auch so durchgeführt werden, dass Code-Knackern die Arbeit wesentlich erschwert wird, da sie nicht im Voraus bestimmen können, welche der extra eingefügten Leerstellen, die im codierten Text vorliegen, wirklich die Daten für den Nachweis der Identität enthalten.

[0007] Es ist daher eine allgemeine Aufgabe der Erfindung, ein Verfahren bereitzustellen, das die zum Nachweis der Identität eines Textdokumentes notwendigen Informationen in Form von zusätzlichen Zwischenwort-Leerzeichen in den Hauptteil des Dokumentes selbst integriert.

[0008] Es ist eine weitere Aufgabe der Erfindung zu ermöglichen, dass der Empfänger des Dokumentes genau dasselbe Format wiederherstellen kann, einschließlich der Zahl der Leerzeichen, wie das des Originaltextes.

[0009] Es ist noch eine weitere Aufgabe der Erfindung, die zusätzlichen Leerzeichen, die tatsächlich die Identitätsnachweisdaten enthalten, mit sinnlosen Leerzeichen zu vermischen, um einen Angreifer noch mehr zu verwirren.

[0010] Weitere Aufgaben, Eigenschaften und Vorteile der vorliegenden Erfindung werden für den Fachmann offensichtlich beim Studium der nachfolgenden Beschreibung mit Bezug auf die beigefügten Zeichnungen. Es ist beabsichtigt, dass sämtliche weitere Vorteile dort eingefügt sind.

Zusammenfassung der Erfindung

[0011] Es wird ein Verfahren zur Markierung eines Original-Textdokumentes beschrieben, wobei das Verfahren darin besteht, die Zahl der vorhandenen Zwischenwort-Leerzeichen des Textes zu verändern. Zunächst wird eine reversible Transformation auf das Original-Textdokument angewendet, um zu erreichen, dass alle Zwischenwort-Intervalle ausschließlich aus einer ungeraden Anzahl von Leerzeichen bestehen. Danach wird der transformierte Originaltext in eine erste und zweite Teilmenge von Wörtern einschließlich deren endseitigen Zwischenwort-Intervalle getrennt. Ein Identitätsnachweismuster, das die Anzahl der Zwischenwort-Intervalle in der ersten Teilmenge unterbringt, wird dann unter Verwendung des Originaltextes und einem Geheim-Schlüssel als Eingabe berechnet. Somit werden Zwischenwort-Leerzeichen an Stellen hinzugefügt, die dem Identitätsnachweismuster entsprechen. Danach wird aus der kanonischen Form (das ist eine Textform, in der alle überschüssigen Zwischenwort-Leerzeichen bis auf eines entfernt sind) der ersten Teilmenge und einem Geheim-Schlüssel ein Verfälschungsmuster berechnet, das auch der Anzahl der Zwischenwort-Intervalle entspricht, so dass die Anzahl der Zwischenwort-Leerzeichen weiter modifiziert wird und das Identitätsnachweismuster, das eben der ersten Teilmenge eingefügt wurde, wird unscharf. Obwohl die zweite Teilmenge das Identitätsnachweismuster nicht enthält, wird auch sie auf ähnliche Weise verfälscht, bevor die erste und zweite Teilmenge zusammengefügt werden und man einen markierter Text erhält, der einem Nachweis der Identität ausgesetzt werden kann.

[0012] Ein Verfahren zum Nachweis der Identität eines gemäß dem hier oben beschriebenen Verfahrens markierten Textdokuments wird ebenfalls beschrieben. Der erste Schritt besteht in der Teilung des markierten Textdokumentes, um die erste und zweite Teilmenge von Wörtern und Intervallen zu erhalten. Danach wird in beiden Teilmengen der Effekt des Verfälschungsmusters entfernt. Das erlaubt auch die Gewinnung des Identitätsnachweismusters, das in die erste Teilmenge eingebettet war, wonach die Teilmengen wieder vereint werden. In diesem Stadium sind alle Zwischenwort-Intervalle wieder ungeradzahlig, und die Transformation, die vom ersten Verfahren angewendet wurde, wird umgekehrt, so dass genau dasselbe Format wie das des Originaltextes wiederhergestellt ist. Weiterhin wird schließlich wie im ersten Verfahren ein Identitätsnachweismuster berechnet, das mit dem weiter oben abgeleiteten Identitätsnachweismuster verglichen wird. Falls diese übereinstimmen, gilt der markierte Text als echt.

[0013] Zudem wird ein System dargelegt, das die Verfahren der Erfindung ausführt. Die Verfahren und das System der Erfindung erlauben es, die Identität des Textdokuments nachzuweisen, wobei das Identitätsnachweismuster eingebettet und tief im Textdokument selbst versteckt ist und das genaue Originalformat einschließlich der Anzahl der Zwischenwort-Leerstellen vom Empfänger wiederhergestellt wird.

Kurze Beschreibung der Zeichnungen

[0014] Fig. 1 zeigt die zur Beschreibung der Erfindung angenommenen Festlegungen.

[0015] Fig. 2 stellt die zur Ausführung der Erfindung verwendete Funktion G dar

[0016] Fig. 3 beschreibt den Gesamtprozess zur Markierung eines Textdokumentes

[0017] Fig. 4 bezieht sich auf die zur Erzeugung ungeradzahligen Zwischenwort-Intervalle verwendete Transformation

[0018] Fig. 5 zeigt, wie das Textdokument in eine erste und zweite Teilmenge aufgespalten wird

[0019] Fig. 6 zeigt den Prozess der Einbettung des Identitätsnachweismusters und Verfälschungsmusters in die erste Teilmenge

[0020] Fig. 7 beschreibt den Gesamtprozess des Nachweises der Identität eines markierten Textdokumentes

Ausführliche Beschreibung der bevorzugten Ausführungsform

[0021] Fig. 1 zeigt, welche Festlegungen im Rest der Beschreibung verwendet werden und was die kanonische Form [**120**] eines Textes im Rahmen der Erfindung ist. Um die Erfindung zu beschreiben, ist ein Text [**100**] dargestellt, der mit einem Begrenzer beginnt und endet, d.h. ein vertikaler Balken [**105**]. Dieser Begrenzer ist nicht Bestandteil des Textes selbst und nur dazu vorhanden, um ihn unzweideutig zuzuordnen. Auf ähnliche Weise werden Wörter durch Leerstellen getrennt, die mit einem Winkelzeichen (caret ^) [**110**] abgebildet werden. Somit besteht ein Text z.B. aus Wörtern [**125**] und Zwischenwort-Intervallen, die mindestens ein Leerzeichen [**110**] umfassen, obwohl mehrere Leerstellen [**115**] auftreten können, worunter die Lesbarkeit jedoch nicht leidet. Die kanonische Form eines Textes [**120**] ist lediglich die Originalform des Textes [**100**], woraus alle überschüssigen Zwischenwort-Leerstellen bis auf eine entfernt wurden.

[0022] Fig. 2 zeigt eine Funktion G [**200**], die zur Ausführung der Erfindung notwendig ist und die auf vielerlei Arten mit Techniken und Verfahren realisiert werden kann, die dem Fachmann auf dem Gebiet bestens bekannt sind. Ungeachtet dessen, wie die Funktion G realisiert wird, wird vorausgesetzt, dass sie eine Ausgabe S [**205**] erzeugen kann, welche von drei Arten von Vorgaben abhängig ist. Zunächst wird S abhängig von einem Eingabetext [**220**] gemacht, der den in Fig. 1 gezeigten ähnelt. Zweitens muss die Ausgabe S auch von einem Schlüssel [**230**] abhängen, der von den am Identitätsnachweisprozess beteiligten Parteien gemeinsam benutzt wird. Drittens ist sie abhängig von einem Parametersatz [**210**], der die Art und Weise festlegt, wie die Funktion G den Eingabetext und insbesondere den Schlüssel zu verarbeiten hat, womit die erwartete Art und das Format der Ausgabe S in einer speziellen Instanz der Funktion festgelegt werden. Als ein Beispiel dafür, wie die Funktion G von der Erfindung eingesetzt wird, ist in Fig. 1 die kanonische Form des Textes [**220**] der Eingabetext, der als ASCII-formatiert vorausgesetzt wird. Der Schlüssel ist z.B. eine alphanumerische Textzeichenfolge [**230**], die geheim gehalten werden muss. Anschließend kann der Parameter [**210**] gesetzt werden, um die Funktion G anzuweisen, beispielsweise einer Zeichenfolge aus 23 Binär-Bits [**215**] zu erzeugen. Der Fachmann wird erkennen, dass die Funktion G so wie hier oben beschrieben realisiert werden kann, z.B. aus einer nicht umkehrbaren Hash-Funktion, die der Erzeugung einer eindeutigen Zusammenfassung des Eingabetextes und des geheimen Schlüssels dient, wiederum abhängig von Eingabeparametern wie die Anzahl der erwarteten Bits, so dass die Ausgabe S zu jedem speziellen Schritt der in den folgenden Figuren beschriebenen Erfindung passend zugeschnitten werden kann. Nicht umkehrbare Hash-Funktionen, die unterschiedlich bezeichnet werden, z.B. mit Kompressions-Funktion, Nachrichtenzusammenfassung, haben großes Interesse erregt und spielen in der modernen Kryptographie eine große Rolle. Eine gute Übersicht über dieses Thema kann in „Applied Cryptography" gefunden werden, ein von Bruce Schneier verfasstes und von John Wiley & Sons veröffentlichtes Buch, zweite Ausgabe 1996. Was die zur Ausführung der Erfindung benötigte Hash-Funktion in Bezug auf die allgemeine Beschreibung auszeichnet und in dem oben genannten Buch und in der zahlreichen Literatur zu diesem Thema beschrieben wird, ist, dass sie zusätzlich zum Eingabetext und insbesondere zum Schlüssel Eingabeparameter annehmen muss, um die Größe der Ausgabe auf eine spezielle Instanz der Funktion anzupassen. Obwohl dies anders als bei Standard-Hashfunktionen ist, die im Allgemeinen eine Zusammenfassung fester Größe für einen verschlüsselten Text erzeugen, entsteht dadurch für den Fachmann kein größeres Problem, so eine Funktion entweder wie oben vorgeschlagen von einer Standard-Hashfunktion aus oder mit einem alternativen Verfahren, das zu einer besonderen Ausführungsform der Erfindung besser passen würde, solch eine Funktion zu entwickeln.

[0023] Fig. 3 zeigt die Hauptschritte des Verfahrens für die Erfindung. Das Verfahren startet mit einem Text [**300**], der zum Nachweis der Identität markiert werden soll. Zuerst wendet man eine reversible Transformation [**305**] an, um alle Zwischenwort-Intervalle aus ungeradzahliger Anzahl von Leerzeichen bestehend zu bekommen. Eine Möglichkeit, dies zu erreichen, besteht darin, zur Zahl N der vorhandenen Leerstellen N-1 zusätzliche Leerstellen hinzuzufügen, so dass im Fall einer einzelnen Zwischenwort-Leerstelle zwischen zwei Wörtern (der gewöhnliche Fall) diese Zahl nach der Transformation nicht

verändert ist. Falls dort jedoch zwei Leerstellen sind, so fügt man zwei minus eine hinzu, d.h. eine zusätzliche Leerstelle zu den beiden vorhandenen, womit ein ungeradzahliges Drei-Leerstellen-Intervall entsteht, usw. Daher ist nach der Erledigung von Schritt [**305**] der „Text" so transformiert, dass er nur aus ungeradzahligen Zwischenwort-Leerstellen besteht. Anschließend wird der transformierte Text in zwei Teilmengen „stext1" und „stext2" aus zufällig ausgewählten Wörtern mit ihren zugehörigen endseitigen Leerstellen aufgeteilt. Dieser Schritt, der in Fig. 5 weiter unten beschrieben ist, wird unter Verwendung der kanonischen Form des Textes [**300**], die im Schritt [**302**] erzeugt wurde, und einem Geheimschlüssel [**312**] als Eingabe über den transformierten, im Schritt [**305**] erhaltenen Text ausgeführt. Der nächste Schritt [**315**] besteht in der Erzeugung eines Binärcodes für den Nachweis der Identität, d.h. ein Binärvektor, dessen Länge der Anzahl der Zwischenwort-Intervalle von „stext1" entspricht. Dies wird erreicht, indem die in Fig. 2 beschriebene Funktion G eingesetzt wird. Es wird ein Code eingesetzt, um mehrere Zwischenwort-Leerstellen einzufügen, beispielsweise an den Stellen, die den Einsen des Binärvektors entsprechen (auch die Nullen können genauso eingesetzt werden), so dass die Anzahlen der Zwischenwort-Leerstellen, die alle ungeradzahlig waren, nun entweder gerad- oder ungeradzahlig sind. An dieser Stelle könnte der Empfänger den Text auf Identität prüfen, um jedoch einem Angreifer die Aufgabe, den Code zu knacken, zu erschweren, wird die Anzahl der Zwischenwort-Intervalle weiter verändert, um das Muster der Zwischenwort-Leerstellen zu verfälschen. Dazu wird ein weiterer Binärvektor im Schritt [**330**] auf ähnliche Weise wie im Schritt [**315**] erzeugt, ausgehend von der kanonischen Form von „stext1", genannt „cstext1" und erzeugt im Schritt [**325**], und einem Geheimschlüssel, um das obige Identitätsnachweismuster von Leerstellen zu verfälschen. Dann wird der binäre Verfälschungsvektor im Schritt [**335**] so eingesetzt, dass für jede Zwischenwort-Position von „stext1", die z.B. einer Eins zugeordnet ist, im Fall einer ungeradzahligen Anzahl von Leerstellen (1, 3, ...) ein zusätzliches Leerzeichen eingefügt wird, im Fall einer geradzahligen Anzahl (2, 4, ...) stattdessen eine Leerstelle entfernt wird. Dies verhindert, dass der Identitätsnachweiscode direkt lesbar ist.

[0024] Was die Schritte des Verfälschens anbelangt, so wird derselbe Prozess auf die zweite Teilmenge „stext2" angewendet. Die Schritte [**345**], [**350**] und [**355**] sind somit identisch mit den äquivalenten Schritten, die eben für „stext1" beschrieben wurden.

[0025] Danach werden die markierten und verfälschten „stext1" und „stext2" auf entgegengesetzte Weise wie im Schritt [**310**], an dem der transformierte, am Schritt [**305**] erhaltene Text aufgeteilt wurde und ein markierter „ftext" erhalten wurde, der auf Identität geprüft werden kann, wieder zusammengeführt [**340**]. Dieser letzte Schritt setzt offenbar voraus, dass auf die Art und Weise, in der die Aufteilung im Schritt [**310**] erfolgte, zurückgegriffen wird, damit die Wörter (und die endseitigen Leerstellen) richtig wieder zusammengeführt werden.

[0026] Fig. 4 veranschaulicht Schritt [**305**] in Fig. 3 genauer, an dem die Anzahl der Zwischenwort-Leerstellen so transformiert wird, dass ausschließlich ungeradzahlige Anzahlen von Leerstellen zwischen je zwei Wörtern entstehen. In diesem Beispiel fügt die Funktion, die zur Transformation des Textes [**420**] in den Text [**425**] verwendet wird, N-1 zusätzliche Leerstellen zu den N bestehenden Leerstellen ein, was bei diesem bestimmten Text [**420**] bedeutet, dass nur an zwei Stellen [**435**] die Anzahl der Leerstellen von zwei auf drei verändert werden.

[0027] Fig. 5 bezieht sich hauptsächlich auf Schritt [**310**], ebenso auf die Schritte [**302**] und [**307**] aus Fig. 3, von wo an der Text [**500**] aufgeteilt wird. Obwohl es viele verschiedene äquivalente Möglichkeiten für diese Schritte gibt, wird ebenso die Funktion G, die in Fig. 2 beschrieben ist, in einer bevorzugten Ausführungsform der Erfindung verwendet. Dazu wird unter Verwendung der kanonischen Form des Textes und des gemeinsam verwendeten Geheimschlüssels als Eingaben die Funktion G so definiert, dass sie einen binären Teilungsvektor [**510**] erzeugt, der der Anzahl der Zwischenwort-Intervalle entspricht.

[0028] Es ist an dieser Stelle erwähnenswert, dass bei jedem zur Aufteilung des Textes eingesetzten Verfahren dieses für eine gegebene Kombination aus „ctext1" und Geheimschlüssel eine eindeutige Art und Weise zur Aufteilung des Textes gewährleisten muss, so dass der Empfänger eines auf Identität überprüften Textes, der gemäß dem Verfahren der Erfindung markiert wurde, beim Empfang dieselbe Aufteilung erhalten kann. In der Praxis erfordert dies, dass in einer bevorzugten Ausführungsform der Erfindung, die die vorher beschriebene Funktion G einsetzt, die zu verwendenden Eingabeparameter im Voraus zwischen dem Absender und dem Empfänger abgesprochen werden (oder das Verfahren, diese eindeutig zu bestimmen).

[0029] Anschließend werden mittels des binären Teilungsvektors [**510**] Wörter und die zugehörigen endseitigen Leerstellen, die dem gesetzten Bit des Vektors entsprechen, als zu einer Teilmenge, z.B. „stext1" [**520**], zugehörig bezeichnet, während die den nicht gesetzten Bits entsprechenden als zur anderen Teilmenge „stext2" [**530**] zugehörig bezeichnet werden. Wie bereits erwähnt wurde, muss auf den geteilten Binärvektor [**510**] zurückgegriffen werden, damit die im Schritt [**340**] von Fig. 3 beschriebene Wiederzusammenführung von Teilmengen ord-

nungsgemäß erfolgen kann.

[0030] Fig. 6 veranschaulicht, wie Zwischenwort-Leerstellen in den Schritten [**320**] und [**335**] von Fig. 3 modifiziert werden, wobei der Identitätsnachweisvektor [**610**], der beim Schritt [**315**] berechnet wurde, und der beim Schritt [**330**] berechnete Verfälschungsvektor [**630**] entsprechend vereint werden. Eine Teilmenge des Textes („stext1"), die nur eine ungeradzahlige Anzahl von Zwischenwort-Leerstellen [**600**] aufweist und welche wie in Fig. 5 erklärt erhalten wurde, wird weiter modifiziert. Zusätzliche Leerstellen werden entsprechend den gesetzten Bits des Identitätsnachweisvektors [**610**] eingefügt und so der Text [**620**] erzeugt. Dieser letztere Text wird wiederum vom Verfälschungsvektor [**630**] verändert, der an den dem gesetzten Bit des Vektors entsprechenden Positionen ein zusätzliches Leerzeichen bei ungerader Anzahl von Leerstellen einfügt und eine Leerstelle entfernt, falls die Anzahl der Leerstellen gerade ist. Das Ergebnis dieser Transformation ist Text [**640**].

[0031] Fig. 7 stellt den Identitätsnachweisprozess dar, der auf einen Text angewendet wird, welcher als gemäß dem gesamten in Fig. 3 beschriebenen Verfahren markiert angenommen wird. Er ist nahezu die Umkehrung dessen, was in dieser letzteren Figur dargestellt ist und erlaubt es, den Text genau wie vom Absender formatiert wiederzugewinnen. Der Prozess startet somit, wenn der auf Identität zu prüfende „ftext" empfangen wird [**700**]. Die kanonische Form dieses Textes wird am Schritt [**702**] erzeugt und „ctext" erhalten, was es erlaubt, in Verbindung mit dem gemeinsam genutzten Geheimschlüssel [**712**] aus Schritt [**707**] „ftext" in eine erste Teilmenge „sftext1" und eine zweite Teilmenge „sftext2" von Wörtern mit endseitigen Leerstellen beim Schritt [**710**] aufzuteilen. Obwohl die Anzahl der endseitigen Leerstellen im Allgemeinen verschieden wäre, was die Wörteraufteilung anbelangt, muss das Ergebnis der Teilungsoperation offenbar übereinstimmen mit dem, was im entsprechenden Schritt [**310**] in Fig. 3 erhalten wurde, sofern der Geheimschlüssel tatsächlich derselbe ist.

[0032] Anschließend besteht der nächste Schritt [**725**] darin, eine kanonische Form von „sftext1", d.h. „csftext1" zu erzeugen, was es in Verbindung mit dem Geheimschlüssel mit der Funktion G [**730**] erlaubt, einen Verfälschungsvektor zu erzeugen, der am nächsten Schritt [**735**] dem Löschen (der Umkehrung) des am Schritt [**335**] von Fig. 3 zum Verstecken des Identitätsnachweis-Codes Erfolgten dient. An dieser Stelle kann der Identitätsnachweisvektor, der vom Verfasser des Textes berechnet wurde, im Schritt [**720**] abgeleitet werden, indem die Anzahl aller Zwischenwort-Leerstellen jeweils in ihre nächstgelegene ungerade Anzahl umgekehrt wird. Dies bedeutet, bei gerader Anzahl eine Leerstelle, bei ungerader Anzahl keine Leerstelle entfernt wird, so dass ein abgeleiteter Identitätsnachweisvektor erhalten wird, dessen gesetzte Bits den Positionen entsprechen, am denen Leerzeichen entfernt werden mussten.

[0033] Ähnlich den Schritten [**725**], [**730**] und [**735**] werden die Schritte [**745**], [**750**] und [**755**] an „sftext2" ausgeführt, um die Auswirkung des Verfälschungsvektors auf die andere Teilmenge ebenso auszulöschen.

[0034] Anschließend werden die beiden Teilmengen wieder zusammengeführt [**740**], um den Text zurückzubekommen, der nur aus ungeradzahligen Intervallen besteht. Daraufhin folgt [**705**] die Anwendung der im Schritt [**305**] verwendeten Umkehr-Transformation, womit der Originaltext wieder erhalten werden kann, d.h. „text" ist genau so wie vom Verfasser formatiert.

[0035] Daher bestehen die letzten Schritte darin, für „text" eine Identitätsprüfung durchzuführen, indem aus ihm und einem Geheimschlüssel ein binärer Identitätsnachweisvektor [**715**] berechnet wird, der beim Vergleich [**760**] mit dem aus Schritt [**720**] sich ergebenden übereinstimmen muss, um „text" auf Identität zu überprüfen. Andernfalls wird der Text als unecht zurückgewiesen.

Patentansprüche

1. Verfahren zur Markierung eines Original-Textdokumentes [**300**], das aus Wörtern [**125**] besteht, die durch Zwischenwort-Intervalle getrennt sind, wobei die Zwischenwort-Intervalle ein [**110**] oder mehrere [**115**] Leerzeichen umfassen, wobei das Verfahren in der Veränderung der Anzahl dieser Leerzeichen besteht und die folgenden Schritte umfasst:
Anwenden einer reversiblen Transformation [**305**] auf das Original-Textdokument, damit alle Zwischenwort-Intervalle ausschließlich aus einer ungeraden Anzahl von Leerzeichen bestehen [**425**];
Aufteilen [**310**] des transformierten Original-Textdokumentes in eine erste Teilmenge [**520**] und eine zweite Teilmenge [**530**] der Wörter einschließlich der endseitigen Zwischenwort-Intervalle dieser Wörter;
und, für die erste Teilmenge: Berechnen [**315**] eines Identitätsprüfungsmusters [**610**] aus dem Original-Textdokument und einem Geheimschlüssel [**312**], das zur Zahl der Intervalle der ersten Teilmenge passt;
Hinzufügen [**320**] von Zwischenwort-Leerzeichen an Stellen, die dem Identitätsprüfungsmuster entsprechen;
Erzeugen [**325**] der kanonischen Form der ersten Teilmenge;
Berechnen [**330**] eines Verfälschungsmusters [**630**] aus der kanonischen Form der ersten Teilmenge und dem Geheimschlüssel, das der Anzahl der Intervalle der ersten Teilmenge entspricht;

Verändern [**335**] der Anzahl der Zwischenwort-Leerzeichen gemäß dem Verfälschungsmuster;
und, für die zweite Teilmenge: Erzeugen [**345**] der kanonischen Form der zweiten Teilmenge;
Berechnen [**350**] eines Verfälschungsmusters aus der kanonischen Form der zweiten Teilmenge und dem Geheimschlüssel, das der Zahl der Intervalle der zweiten Teilmenge entspricht;
Verändern [**355**] der Anzahl der Zwischenwort-Leerzeichen gemäß dem Verfälschungsmuster;
Wiederzusammenführen [**340**] der ersten und zweiten Teilmenge, um einen für den Nachweis der Identität markierten Text zu erhalten.

2. Verfahren zum Nachweis der Identität eines markierten Textdokumentes [**700**], wobei das Textdokument aus Wörtern [**125]** besteht, die durch Zwischenwort-Intervalle getrennt sind, wobei die Zwischenwort-Intervalle ein [**110**] oder mehrere [**115**] Leerzeichen umfassen, wobei das Verfahren in der Prüfung der Anzahl der Leerzeichen besteht und das Verfahren die folgenden Schritte umfasst:
Aufteilung des markierten Textdokumentes [**710**] in eine erste und zweite Teilmenge von Wörtern "einschließlich der Zwischenwort-Intervalle am Wortende;
und für die erste Teilmenge:
Erzeugen [**725**] einer kanonischen Form der ersten Teilmenge;
Berechnen [**730**] eines Verfälschungsmusters aus der kanonischen Form der ersten Teilmenge und einem Geheimschlüssel [**712**], das der Anzahl der Intervalle der ersten Teilmenge entspricht;
Löschen [**735**] von Veränderungen, die das Verfälschungsmuster für die Anzahl der Zwischenwort-Leerzeichen bewirkt hat;
dadurch Ableiten [**720**] eines Identitätsprüfungsmusters, womit in allen Zwischenwort-Intervallen eine ungerade Anzahl von Leerzeichen entsteht;
und für die zweite Teilmenge:
Erzeugen [**745**] einer kanonischen Form der zweiten Teilmenge;
Berechnen [**750**] eines Verfälschungsmusters aus der kanonischen Form der zweiten Teilmenge und dem Geheimschlüssel [**712**], das der Anzahl der Intervalle der zweiten Teilmenge entspricht;
Löschen [**755**] der Veränderungen, die das Verfälschungsmuster für die Anzahl der Zwischenwort-Leerzeichen bewirkt hat, womit in allen Zwischenwort-Intervallen eine ungerade Anzahl von Leerzeichen entsteht;
Wiederzusammenführen der ersten und zweiten Teilmenge;
Anwenden einer Umkehrtransformation [**705**], um das Original-Textdokument wiederzugewinnen;
Berechnen [**715**] eines Identitätsprüfungsmusters aus dem wiedergewonnenen Original-Textdokument und dem Geheimschlüssel [**712**], das der Anzahl der Intervalle des wiedergewonnenen Original-Textdokumentes entspricht;
Vergleichen [**760**] des abgeleiteten Identitätsprüfungsmusters [**720**] und des berechneten Identitätsprüfungsmusters [**715**];
falls beide genau übereinstimmen:
Annehmen des markierten Textdokumentes als echt;
Falls nicht:
Zurückweisen des markierten Textdokumentes als gefälscht.

3. Verfahren nach einem der vorangehenden Ansprüche, wobei die Aufteilungsschritte [**310**] [**710**] folgende Vorab-Schritte beinhaltet:
Erzeugen [**302**] [**702**] einer kanonischen Form eines Textdokumentes;
Berechnen [**307**] [**707**] eines Aufteilungsmusters [**510**] aus der kanonischen Form des Textdokumentes und dem Geheimschlüssel, das der Anzahl der Intervalle des Textdokumentes entspricht;
dadurch Ermöglichen, dass das Textdokument [**300**] [**700**] anhand der gesetzten und nicht gesetzten Bits des Teilungsmusters aufgeteilt und wieder zusammengeführt wird.

4. Verfahren nach einem der vorhergehenden Ansprüche, wobei das Identitätsnachweismuster [**610**], das Verfälschungsmuster [**630**] und das Teilungsmuster [**510**] binäre Vektoren sind, die aus einer Anzahl von Bits bestehen, die der Anzahl der zugehörigen Zwischenwort-Intervalle entspricht.

5. Verfahren nach einem der vorhergehenden Ansprüche, wobei die kanonische Form [**120**] entsteht, indem alle Leerzeichen der Zwischenwort-Intervalle bis auf eines entfernt werden.

6. Verfahren nach einem der vorhergehenden Ansprüche, worin die Änderungsschritte Folgendes beinhalten:
an den Stellen, die den gesetzten Bits der Verfälschungsmusters entsprechen:
Hinzufügen eines Leerzeichens, falls die Zwischenwort-Intervalle aus einer ungeraden Anzahl von Leerzeichen bestehen.
Entfernen eines Leerzeichens, falls die Zwischenwort-Intervalle aus einer geraden Anzahl von Leerzeichen bestehen.

7. Verfahren nach einem der vorhergehenden Ansprüche, worin die Ändern- und Löschen-Schritte gleichartig ablaufen.

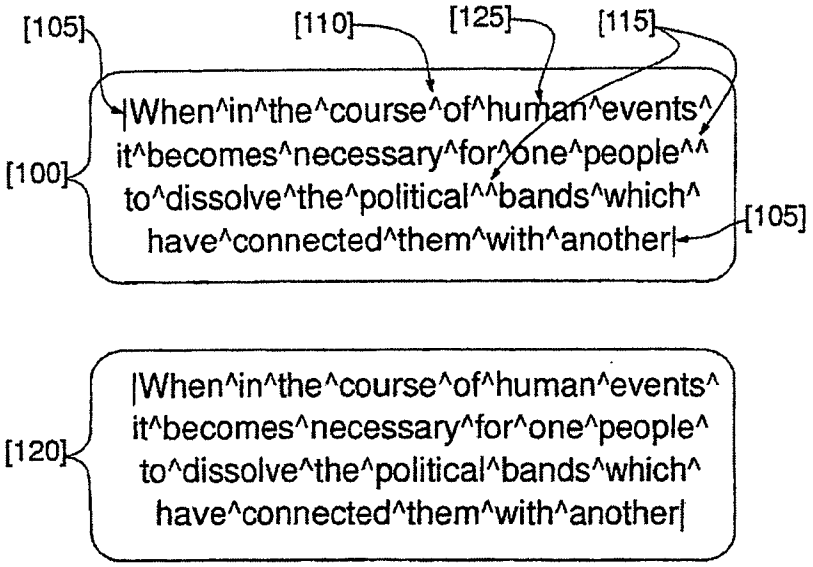
8. Verfahren nach einem der vorhergehenden Ansprüche, worin die Ableitungsschritte umfassen:
Entfernen eines Leerzeichens in diesen Zwischenwort-Intervallen, die aus einer geraden Anzahl von Leerzeichen bestehen;
Erhalten eines binären Identitätsnachweisvektors mit gesetzten Bits an Stellen, an denen Leerzeichen entfernt wurden.

9. Identitätsnachweissystem, insbesondere ein System für den Nachweis der Identität eines Textdokumentes, das Mittel umfasst, die zur Ausführung des Verfahrens nach einem der vorherigen Ansprüche in der Lage ist.

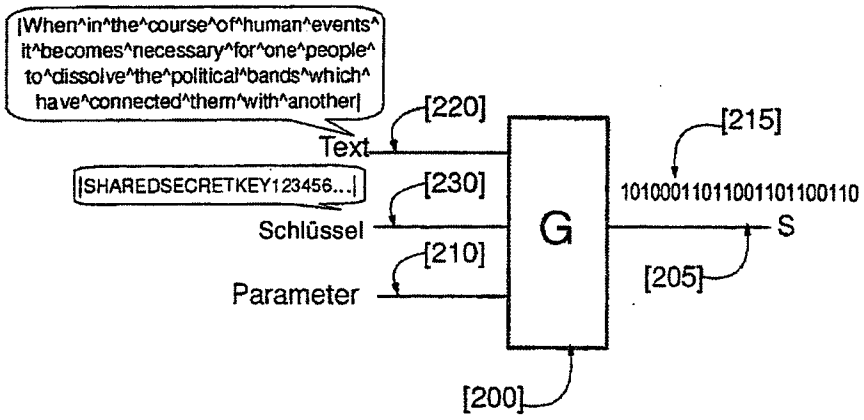
10. Von Computern oder dergleichen lesbares Medium, das Anweisungen zur Ausführung des Verfahrens nach einem der vorhergehenden Ansprüche 1 bis 8 umfasst.

Es folgen 6 Blatt Zeichnungen

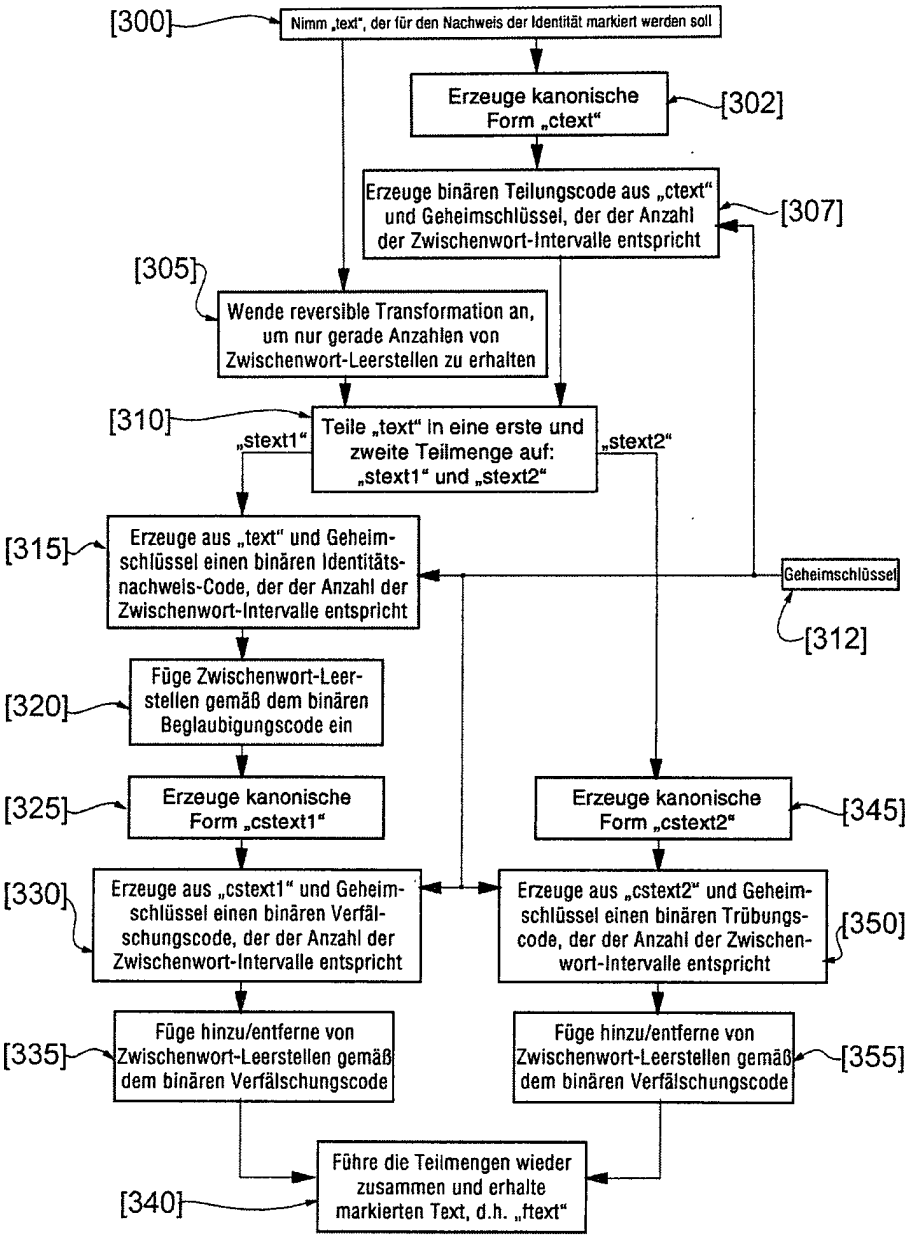
Anhängende Zeichnungen



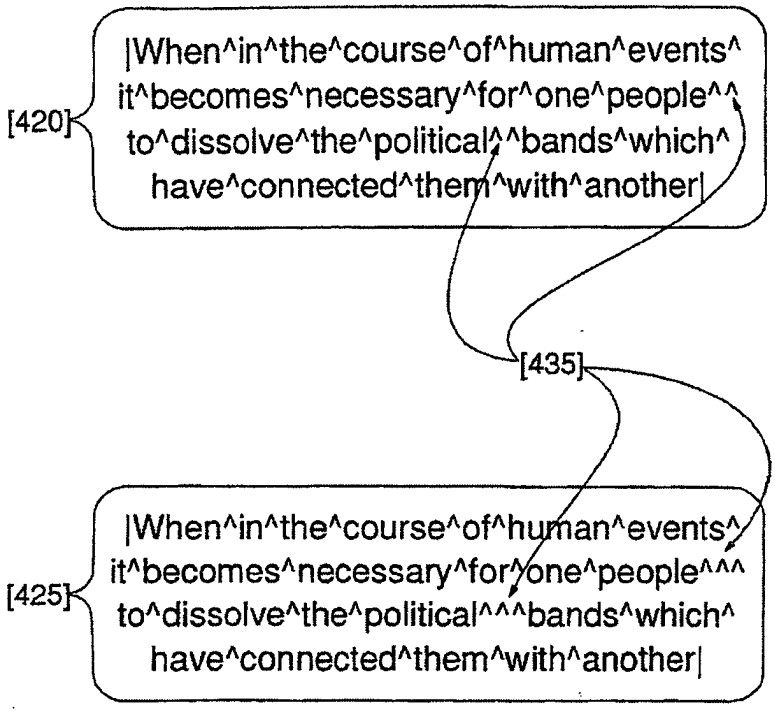
Figur 1



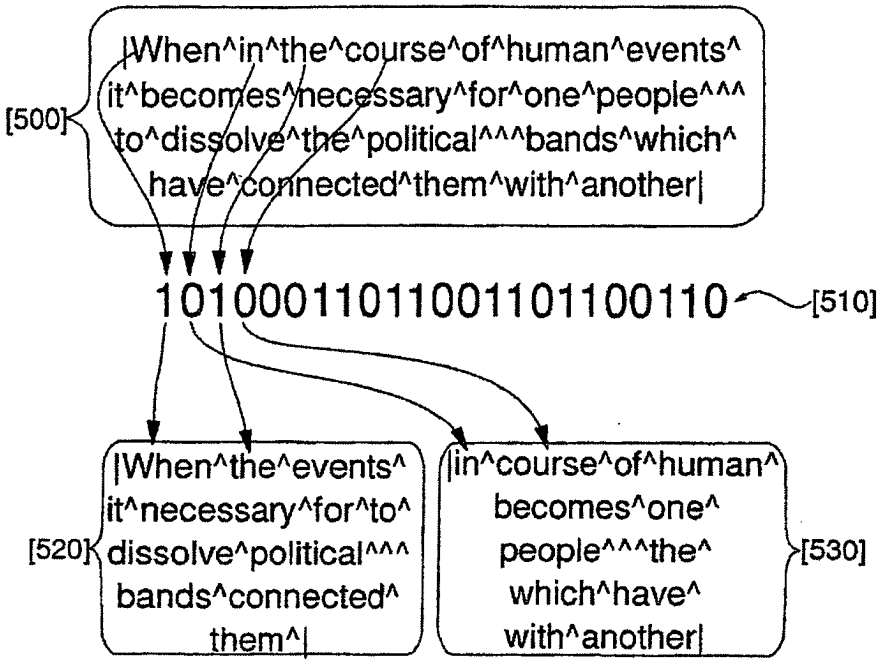
Figur 2


[300]
Nimm „text", der für den Nachweis der Identität markiert werden soll
Erzeuge kanonische Form „ctext"
[302]
Erzeuge binären Teilungscode aus „ctext" und Geheimschlüssel, der der Anzahl der Zwischenwort-Intervalle entspricht
[307]
[305]
Wende reversible Transformation an, um nur gerade Anzahlen von Zwischenwort-Leerstellen zu erhalten
[310]
„stext1"
Teile „text" in eine erste und zweite Teilmenge auf: „stext1" und „stext2"
„stext2"
[315]
Erzeuge aus „text" und Geheimschlüssel einen binären Identitätsnachweis-Code, der der Anzahl der Zwischenwort-Intervalle entspricht
Geheimschlüssel
[312]
[320]
Füge Zwischenwort-Leerstellen gemäß dem binären Beglaubigungscode ein
[325]
Erzeuge kanonische Form „cstext1"
Erzeuge kanonische Form „cstext2"
[345]
[330]
Erzeuge aus „cstext1" und Geheimschlüssel einen binären Verfälschungscode, der der Anzahl der Zwischenwort-Intervalle entspricht
Erzeuge aus „cstext2" und Geheimschlüssel einen binären Trübungscode, der der Anzahl der Zwischenwort-Intervalle entspricht
[350]
[335]
Füge hinzu/entferne von Zwischenwort-Leerstellen gemäß dem binären Verfälschungscode
Füge hinzu/entferne von Zwischenwort-Leerstellen gemäß dem binären Verfälschungscode
[355]
[340]
Führe die Teilmengen wieder zusammen und erhalte markierten Text, d.h. „ftext"

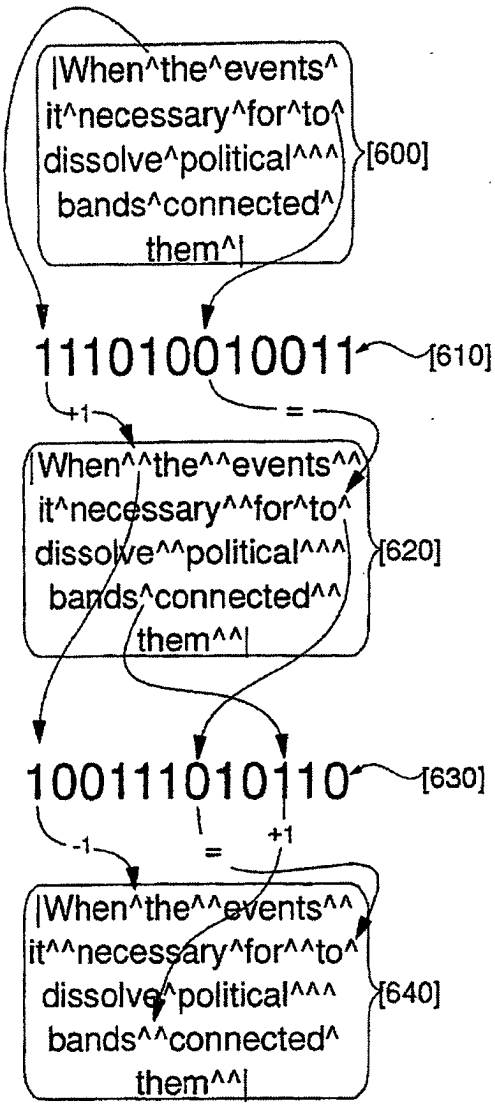
Figur 3


[420]
|When^in^the^course^of^human^events^
it^becomes^necessary^for^one^people^^
to^dissolve^the^political^^bands^which^
have^connected^them^with^another|
[435]
[425]
|When^in^the^course^of^human^events^
it^becomes^necessary^for^one^people^^^
to^dissolve^the^political^^^bands^which^
have^connected^them^with^another|

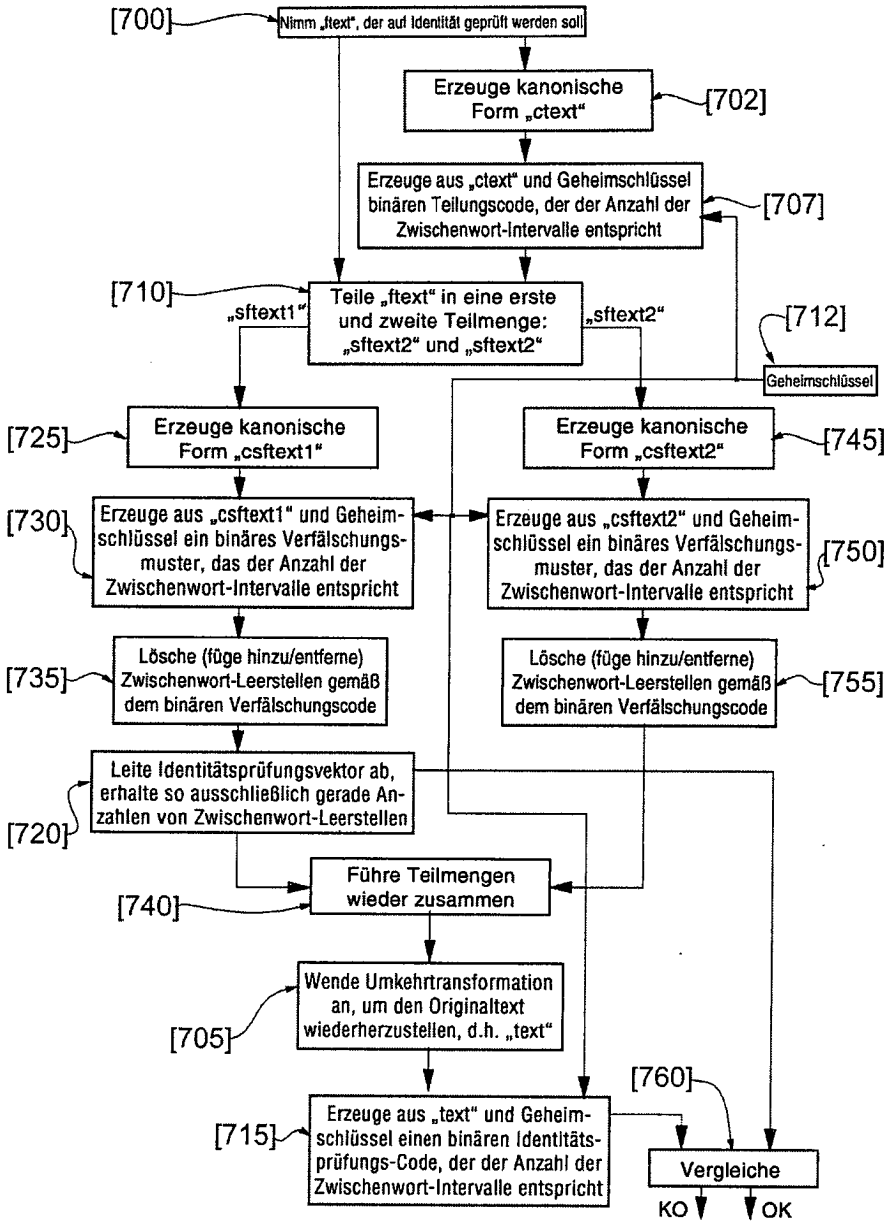
Figur 4


|When^in^the^course^of^human^events^
it^becomes^necessary^for^one^people^^^
to^dissolve^the^political^^^bands^which^
have^connected^them^with^another|
[500]
10100011011001101100110
[510]
|When^the^events^
it^necessary^for^to^
dissolve^political^^^
bands^connected^
them^|
[520]
|in^course^of^human^
becomes^one^
people^^^the^
which^have^
with^another|
[530]

Figur 5


|When^the^events^
it^necessary^for^to^
dissolve^political^^^
bands^connected^
them^|
[600]
111010010011
[610]
+1
=
|When^^the^^events^^
it^necessary^^for^to^
dissolve^^political^^^
bands^connected^^
them^^|
[620]
100111010110
[630]
-1
=
+1
|When^the^^events^^
it^^necessary^for^^to^
dissolve^political^^^
bands^^connected^
them^^|
[640]

Figur 6


[700]
Nimm „ftext", der auf Identität geprüft werden soll
Erzeuge kanonische Form „ctext"
[702]
Erzeuge aus „ctext" und Geheimschlüssel binären Teilungscode, der der Anzahl der Zwischenwort-Intervalle entspricht
[707]
[710]
Teile „ftext" in eine erste und zweite Teilmenge: „sftext2" und „sftext2"
„sftext1"
„sftext2"
[712]
Geheimschlüssel
[725]
Erzeuge kanonische Form „csftext1"
Erzeuge kanonische Form „csftext2"
[745]
[730]
Erzeuge aus „csftext1" und Geheimschlüssel ein binäres Verfälschungsmuster, das der Anzahl der Zwischenwort-Intervalle entspricht
Erzeuge aus „csftext2" und Geheimschlüssel ein binäres Verfälschungsmuster, das der Anzahl der Zwischenwort-Intervalle entspricht
[750]
[735]
Lösche (füge hinzu/entferne) Zwischenwort-Leerstellen gemäß dem binären Verfälschungscode
Lösche (füge hinzu/entferne) Zwischenwort-Leerstellen gemäß dem binären Verfälschungscode
[755]
Leite Identitätsprüfungsvektor ab, erhalte so ausschließlich gerade Anzahlen von Zwischenwort-Leerstellen
[720]
Führe Teilmengen wieder zusammen
[740]
Wende Umkehrtransformation an, um den Originaltext wiederherzustellen, d.h. „text"
[705]
[760]
[715]
Erzeuge aus „text" und Geheimschlüssel einen binären Identitätsprüfungs-Code, der der Anzahl der Zwischenwort-Intervalle entspricht
Vergleiche
KO
OK

Figur 7