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

(10) **DE 603 14 062 T2** 2007.11.22

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 396 978 B1**
(21) Deutsches Aktenzeichen: **603 14 062.9**
(96) Europäisches Aktenzeichen: **03 020 076.0**
(96) Europäischer Anmeldetag: **04.09.2003**
(97) Erstveröffentlichung durch das EPA: **10.03.2004**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **30.05.2007**
(47) Veröffentlichungstag im Patentblatt: **22.11.2007**

(51) Int Cl.[8]: ***H04L 29/06*** (2006.01)

(30) Unionspriorität:
235587 **04.09.2002** **US**

(84) Benannte Vertragsstaaten:
AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LI, LU, MC, NL, PT, RO, SE, SI, SK, TR

(73) Patentinhaber:
Microsoft Corp., Redmond, Wash., US

(74) Vertreter:
Grünecker, Kinkeldey, Stockmair & Schwanhäusser, 80538 München

(72) Erfinder:
Adent, Daniel, Bellevue Washington 98007, US; West, Cory, Central Piont Oregon 97502, US; Dublish, Pratul, Sammamish Washington 98075, US; Strom, Clifford P., Sammamish Washington 98074, US; Crites, Brian D., Sammamish, Washington 98074, US

(54) Bezeichnung: **Schutz für Header-Objekte in Datenströmen**

Beschreibung

Gebiet der Erfindung

[0001] Die vorliegende Erfindung bezieht sich im Allgemeinen auf Datenverifizierung und im Speziellen auf ein Header-Objekt (Kopfobjekt) für eine Datendatei.

Hintergrund der Erfindung

[0002] Normalerweise schließen einige Datendateien und Datenstromformate Header-Objekte ein. Die Header-Objekte schließen "Meta-Inhalt"-Informationen (meta content information) ein, die zur Identifizierung und Verwendung der Inhaltsdaten, die in der Datendatei oder dem Datenstrom eingeschlossen sind, verwendet werden.

[0003] Zum Beispiel ist ein Datenstromformat das verbesserte Streaming-Format (Advanced Streaming Format – ASF), welches ein erweiterbares Dateiformat ist, das zum Speichern aufeinander abgestimmter Multimediadaten entwickelt wurde. Die derzeitige Spezifikation für dieses Format ist unter www.microsoft.com verfügbar. ASF unterstützt die Datenlieferung über eine breite Vielfalt an Netzwerken und Protokollen, während es ein lokales Abspielen erlaubt.

[0004] Jede ASF-Datei setzt sich aus einem oder mehreren Medienströmen zusammen. Das Header-Objekt bestimmt die Eigenschaften der gesamten Datei zusammen mit Datenstrom-spezifischen Eigenschaften. In ASF muss jede Datei ein Header-Objekt haben. Das Header-Objekt bietet am Anfang der ASF-Dateien eine wohl bekannte Byte-Sequenz (die Header-Objekt-GUID (globally unique identifier – globaler eindeutiger Indentifizierer)) und enthält alle Informationen, die benötigt werden, um die Multimediadaten richtig zu interpretieren. Das Header-Objekt kann als ein Container gedacht werden, der Header-Objektinformationen und eine Kombination von Header-Unterobjekten enthält. Die Header-Objektinformationen bestehen aus einem GUID für das Header-Objekt ("ASF_Header_Object"), der Größe des Header-Objekts und der Anzahl an Header-Unterobjekten, die in dem Header-Objekt enthalten sind. Jedes Header-Objekt fängt mit einer GUID an.

[0005] Header-Unterobjekte schließen ein:

- ein Dateieigenschaftsunterobjekt, welches die globalen Charakteristiken der Multimediadaten in der Datei definiert;
- ein Datenstromeigenschaftenunterobjekt, welches die bestimmten Eigenschaften und Charakteristiken eines Mediendatenstroms definiert;
- das Header-Erweiterungsunterobjekt, welches das Hinzuzufügen zusätzlicher Funktionalität zu einer ASF-Datei erlaubt, während eine Rückwärtskompatibilität aufrecht erhalten wird, und ein Container ist, der erweiterte Header-Unterobjekte enthält;
- das Codec-Listenunterobjekt, welches benutzerfreundliche Informationen über die Codecs und Formate bereitstellt, die zum Codieren des Inhalts, der in der ASF-Datei gefunden wird, verwendet werden;
- das Skript-Befehlsunterobjekt, welches eine Liste mit Typ-/Parameter-Paaren aus Unicode-Zeichenfolgen bereitstellt, die zu der Zeitachse der ASF-Datei synchronisiert sind;
- Das Markierungszeichenunterobjekt, welches einen kleinen spezialisierten Index enthält, der verwendet wird, um benannte Sprungpunkte (jump points) innerhalb einer Datei bereit zu stellen, um es einem Inhaltsautor zu erlauben, den Inhalt in logische Abschnitte zu unterteilen, wie zum Beispiel Liedabgrenzungen in einer gesamten CD oder Themenveränderungen während einer langen Präsentation, und um, für die Verwendung durch den Benutzer, jedem Abschnitt einer Datei einen vom Menschen lesbaren Namen zuzuweisen;
- das gegenseitige Bitratenausschlussunterobjekt (bitrate mutual exclusion subobject), welches Videodatenströme identifiziert, die eine gegenseitige Ausschlussbeziehung zueinander haben (mit anderen Worten, nur einer der Datenströme innerhalb solch einer Beziehung kann herunter geladen (streamed) werden und die Reste werden ignoriert);
- das Fehlerkorrekturunterobjekt, welches das Fehlerkorrekturverfahren definiert und Informationen bereit stellt, die durch die Fehlerkorrekturmaschine (engine) zur Wiederherstellung benötigt werden;
- das Inhaltbeschreibungsunterobjekt, welches Autoren erlaubt, gut bekannte Daten, die die Datei und ihre Inhalte beschreiben, aufzuzeichnen, einschließlich Titel, Autor, Copyright, Beschreibung und Bewertungsinformation;
- das erweiterte Inhaltsbeschreibungsunterobjekt, welches Autoren erlaubt, Daten, die die Datei und ihre Inhalte beschreiben, die über die normalen bibliographischen Informationen, wie zum Beispiel, Autor, Copyright, Beschreibung oder Bewertungsinformation hinaus gehen, aufzuzeichnen;
- das Inhalteverschlüsselungsunterobjekt, welches identifiziert, ob der Inhalt durch ein digitales Rechtemanagementsystem (DRM-System) geschützt ist. Dieses Unterobjekt schließt die DRM-Lizenzbeschaffungs-URL, die DRM-Schlüssel-ID und andere DRM-bezogene Meta-Daten ein.
- Das Datenstrom-Bitrateneigenschaftenunterobjekt, welches die durchschnittliche Bitrate von jedem Mediendatenstrom in den Multimediadaten definiert;
- ein Füllunterobjekt (padding subobject), welches ein Blind-Unterobjekt (dummy sub-object) ist, das

verwendet wird, um die Größe des Header-Objekts auszufüllen.

[0006] Die Einheit, die die Datenstromdatei zuerst erzeugt, und irgendwelche nachfolgenden Einheiten, die darauf einwirken, können der Header-Datei Elemente hinzufügen oder sie verändern. Zum Beispiel kann eine inhalterzeugende Einheit eine Datenstromdatei erzeugen und Informationen in das Inhaltbeschreibungsobjekt bezüglich des Inhalts einschließen. Eine zweite Einheit kann Markierungsstellen innerhalb der Daten erzeugen und wünschen, ein Markierungsobjekt mit Spurinformationen (track information) hinzuzufügen. Und eine dritte Einheit, welche die Datenstromdatei vertreibt, kann ein Skript-Befehlsobjekt, das Vorgänge oder Daten für Skripte enthält, hinzufügen. Zum Beispiel kann ein Skript-Befehlsobjekt Informationen enthalten, die ein Webbrowserfenster zu einer bestimmten URL (uniform resource locator – Internetadresse) öffnet.

[0007] Weil eine Anzahl von Einheiten auf eine ASF-Datei einwirken kann, gibt es keine Möglichkeit zu ermitteln, welche Einheit welchen Teil des Header-Objekts erzeugt hat. Zusätzlich kann eine Veränderung der Information durch einen Angreifer (attacker) nicht identifiziert werden.

[0008] Eastlake D. et al., "(Extensible Markup Language) XML – Signature Syntax and Processing", RFC 3275, März 2003, spezifiziert digitale Signatur-Verarbeitungsregeln und eine -Syntax für XML (Extensible Markup Language). Digitale XML-Signaturen werden darin durch das Signaturelement dargestellt, dessen Struktur das erforderliche SignedInfo-Element und einen SignaturValue (Signaturwert) einschließt. Das SignedInfo-Element ist die Information, die tatsächlich signiert wird. Das SignedInfo-Element schließt mindestens ein Referenzelement ein, welches des Weiteren das Extrahierungsverfahren (digest method) und den resultierenden Extrahierungswert (digest value), der über ein identifiziertes Datenobjekt berechnet wird, einschließt. Ein anderes Element, das in dem SignedInfo-Element eingeschlossen ist, ist das SignaturMethod (Signaturverfahren), welches der Algorithmus ist, der verwendet wurde, um das kanonisierte (canonicalized) SignedInfo in das SignatureValue zu konvertieren. Deshalb wird eine XML-Signatur auf einen beliebigen digitalen Inhalt (Datenobjekte) über einen Umweg angewandt. Datenobjekte werden kurzgefasst (digested), der resultierende Wert wird in ein Element zusammen mit anderen Informationen platziert und das Element wird dann kurzgefasst (digested) und kryptographisch signiert. Die Kernvalidierung schließt deshalb eine Referenzvalidierung und die kryptographische Signaturvalidierung der Signatur, die über SignedInfo berechnet wurde, ein.

Kurzfassung der Erfindung

[0009] Es ist die Aufgabe der Erfindung Sicherheit und Flexibilität von Datenstromdateien bereitzustellen und das Entfernen oder Verändern von Daten zu vermeiden.

[0010] Diese Aufgabe wird durch die Erfindung wie in den unabhängigen Ansprüchen beansprucht gelöst.

[0011] Bevorzugte Ausführungsformen sind durch die abhängigen Ansprüche definiert.

[0012] Die vorliegende Erfindung ist auf ein System, Verfahren und Datenstruktur für die Verifizierung von Unterobjekten in einem Header-Objekt gerichtet. Die Erfindung gewährleistet durch eine Einheit eine Verifizierung von einem oder mehreren Unterobjekten in dem Header-Objekt, während sie weiterhin das Ändern der Reihenfolge der Unterobjekte erlaubt. Neue Unterobjekte können ebenso anschließend durch eine andere Einheit erzeugt und verifiziert werden. Die Verifizierung von zwei oder mehr Unterobjekten durch eine vertrauenswürdige Einheit kann kombiniert werden, so dass ein Angreifer Daten nicht entfernen oder verändern kann, wobei es ein Unterobjekt verifizierbar lässt, indem es durch die vertrauenswürdige Einheit signiert worden ist, während das andere Unterobjekt nicht verifizierbar ist.

[0013] Zusätzliche Merkmale und Vorteile der Erfindung werden in der unten liegenden Beschreibung dargelegt.

Kurze Beschreibung der Figuren

[0014] Fig. 1 ist ein Diagramm, das einen Überblick über ein Computersystem darstellt.

[0015] Fig. 2 ist ein Blockdiagramm, das eine Datei gemäß der Erfindung darstellt.

[0016] Fig. 3 stellt den Prozess des Erzeugens eines digitalen Signaturunterobjekts gemäß der Erfindung dar.

[0017] Fig. 4 stellt den Prozess des Verifizierens eines digitalen Signaturunterobjekts gemäß der Erfindung dar.

[0018] Fig. 5 stellt ein digitales Signaturunterobjekt gemäß der Erfindung dar.

Detaillierte Beschreibung der bevorzugten Ausführungsformen:

Überblick

[0019] Ein oder mehrere digitale Signaturunterob-

jekte können erzeugt werden und in das Header-Objekt einer Datendatei platziert werden, um Signaturinformation für Unterobjekte und Regionen von Unterobjekten in dem Header-Objekt vorzusehen. Wenn ein digitales Signaturunterobjekt vorhanden und gültig ist, kann jegliche Bearbeitung oder Verfälschung in den signierten Unterobjekten erkannt werden. Eine Reihenfolge der Unterobjekte braucht nicht bewahrt zu werden.

[0020] Das digitale Signaturunterobjekt enthält ein Array mit Regionsspezifizierern. Jeder Regionsspezifizierer identifiziert eine bestimmte Region innerhalb eines Unterobjekts. Ein Regionsspezifizierer kann ebenso ein gesamtes Unterobjekt identifizieren.

[0021] Das digitale Signaturunterobjekt enthält ebenfalls eine Signatur. Die Signatur ist eine digitale Signatur der Regionen, die in dem Array mit Regionsspezifizierern aufgelistet sind. Die Signatur kann verwendet werden, um zu verifizieren, dass die Regionen, die in dem Regionsspezifizierer-Array aufgelistet sind, nicht verfälscht worden sind.

Beispielhafte Computerumgebung

[0022] Fig. 1 stellt ein Beispiel einer geeigneten Computersystemumgebung **100** dar, in der die Erfindung implementiert werden kann. Die Computersystemumgebung **100** ist nur ein Beispiel einer geeigneten Computerumgebung, und ist nicht gedacht, irgendwelche Einschränkungen bezüglich des Umfangs der Verwendung oder Funktionalität der Erfindung vorzuschlagen. Noch sollte die Computerumgebung **100** interpretiert werden, als hätte sie irgendwelche Abhängigkeiten oder Erfordernisse mit Bezug auf irgendeine oder eine Kombinationen von Komponenten, die in der exemplarischen Arbeitsumgebung **100** dargestellt sind.

[0023] Der gewöhnliche Fachmann kann verstehen, dass ein Computer oder ein anderes Client- oder Server-Gerät als Teil eines Computernetzwerkes oder in einer verteilten Computerumgebung eingesetzt werden kann. In diesem Zusammenhang betrifft die vorliegende Erfindung irgendein Computersystem, das irgendeine Anzahl von Speicher- oder Datenspeichereinheiten und irgendeine Anzahl von Anbindungen und Prozessen, die über irgendeine Anzahl von Speichereinheiten oder Volumina, die in Verbindung mit der vorliegenden Erfindung verwendet werden können, aufweist. Die vorliegende Erfindung kann auf eine Umgebung mit Servercomputern und Clientcomputern angewandt werden, die in einer Netzwerkumgebung oder verteilten Computerumgebung, die Remote- oder lokalen Speicher aufweist, eingesetzt werden. Die vorliegende Erfindung kann ebenso auf allein stehende Computergeräte angewandt werden, die Programmiersprachenfunktionalität, -interpretation und Ausführungsfähigkeiten zum Erzeugen, Empfangen und Übermitteln von Informationen in Verbindung mit Remote- oder lokalen Services aufweisen.

[0024] Die Erfindung ist mit einer Vielzahl von Allzweck- oder Spezialzweckcomputersystemumgebungen oder -konfigurationen betriebsbereit. Beispiele von gut bekannten Computersystemen, Umgebungen und/oder Konfigurationen, die zur Verwendung mit der Erfindung geeignet sein können, schließen ein, sind aber nicht darauf begrenzt, Personalcomputer, Servercomputer, tragbare oder Laptop-Geräte, Multiprozessorsysteme, Mikroprozessor-basierte Systeme, Set Top Boxen, programmierbare Unterhaltungselektroniken, Netzwerk-PCs, Minicomputer, Mainframe-Computer, verteilte Computerumgebungen, die irgendeines der oben genannten Systeme oder Geräte einschließen, und ähnliches.

[0025] Die Erfindung kann im allgemeinen Kontext von computerausführbaren Instruktionen beschrieben werden, wie zum Beispiel Programmmodulen, die durch einen Computer ausgeführt werden. Im Allgemeinen schließen Programmmodule Routinen, Programme, Objekte, Komponenten, Datenstrukturen, etc. ein, die bestimmte Aufgaben ausführen oder bestimmte abstrakte Datentypen implementieren. Die Erfindung kann ebenso in verteilten Computerumgebungen praktiziert werden, wo Aufgaben durch remoteverarbeitende Geräte, die durch ein Kommunikationsnetzwerk oder anderes Datenübertragungsmedium miteinander verbunden sind, ausgeführt werden. In einer verteilten Computer-Umgebung können Programmmodule und andere Daten sowohl in lokalen als auch Remote-Computerspeicherdatenträgern, einschließlich Datenspeichergeräten, liegen. Verteilte Computer ermöglichen die Mitbenutzung von Computerressourcen und Services durch direkten Austausch zwischen den Computergeräten und Systemen. Diese Ressourcen und Services schließen den Austausch von Informationen, Cache-Speicher und Disk-Speicher für Dateien ein. Verteilte Computer ziehen den Vorteil aus Netzwerkkonnektivität, die es Clients erlaubt, ihre gemeinsame Stärke wirksam einzusetzen, um dem gesamten Unternehmen zu nutzen. Diesbezüglich kann eine Vielfalt von Geräten Anwendungen, Objekte oder Ressourcen aufweisen, die die Techniken der vorliegenden Erfindung benutzen.

[0026] Mit Bezug auf Fig. 1 schließt ein exemplarisches System zum Implementieren der Erfindung ein Allzweckcomputergerät in der Form eines Computers **110** ein. Komponenten des Computers **110** können einschließen, sind aber nicht darauf begrenzt, eine Prozessoreinheit **120**, einen Systemspeicher **130** und einen Systembus **121**, der verschiedene Systemkomponenten koppelt, einschließlich dem Systemspeicher mit der Prozessoreinheit **120**. Der Systembus **121** kann irgendeiner von verschiedenen Busstrukturtypen sein, einschließlich einem

Speicherbus oder Speicherkontroller, einem Peripheriebus und einem lokalen Bus, der irgendeine von einer Vielfalt von Busarchitekturen verwendet. Als Beispiel, und nicht Einschränkung, schließen solche Architekturen Industry-Standard-Architecture-Bus (ISA-Bus), Micro-Channel-Architecture-Bus (MCA-Bus), Enhanced-ISA-Bus (EISA-Bus), Video-Electronics-Standards-Association-Local-Bus (VE-SA-local-Bus) und Peripheral-Component-Interconnect-Bus (PCI-Bus) (ebenso als Mezzanine-Bus bekannt) ein.

[0027] Der Computer **110** schließt üblicherweise eine Vielfalt von computerlesbaren Datenträgern ein. Die computerlesbaren Datenträger können irgendwelche von verfügbaren Datenträgern sein, auf die durch den Computer **110** zugegriffen werden kann, und die sowohl flüchtige als auch nicht-flüchtige Datenträger, entfernbare und nicht-entfernbare Datenträger einschließen. Als Beispiel, und nicht Einschränkung, können solche computerlesbaren Datenträger Computerspeicherdatenträger und Kommunikationsdatenträger umfassen. Computerspeicherdatenträger schließen sowohl flüchtige als auch nicht-flüchtige, entfernbare als auch nicht-entfernbare Datenträger ein, die in irgendeinem Verfahren oder einer Technologie zur Speicherung von Informationen implementiert sind, wie zum Beispiel computerlesbare Instruktionen, Datenstrukturen, Programmmodule oder andere Daten. Computerspeicherdatenträger schließen ein, sind aber nicht darauf begrenzt, RAM, ROM, EEPROM, Flash-Speicher oder eine andere Speichertechnologie, CD-ROM, digital-versatile-disks (DVD) oder andere optische Disk-Speicher, magnetische Kassetten, magnetische Bänder, magnetisch Disk-Speicher oder andere magnetische Speichergeräte oder irgendeinen anderen Datenträger, der verwendet werden kann, um die gewünschten Informationen zu speichern und auf den durch Computer **110** zugegriffen werden kann. Kommunikationsdatenträger enthalten üblicherweise computerlesbare Instruktionen, Datenstrukturen, Programmmodule oder andere Daten in einem modulierten Datensignal, wie zum Beispiel einer Trägerwelle oder anderem Transportmechanismus und schließen irgendwelche Informationslieferdatenträger ein. Der Begriff "moduliertes Datensignal" meint ein Signal, das eine oder mehrere seiner Charakteristiken gesetzt oder in solch einer Weise verändert hat, um Informationen in dem Signal zu kodieren. Als Beispiel und nicht Einschränkung, schließen Kommunikationsdatenträger verkabelte Datenträger, wie zum Beispiel ein verkabeltes Netzwerk oder eine direktverkabelte Verbindung, und kabellose Datenträger, wie zum Beispiel akustische, RF-, infrarote oder andere kabellose Datenträger ein. Kombinationen von irgendwelchen der oben genannten sollten ebenso in dem Umfang der computerlesbaren Datenträger eingeschlossen sein.

[0028] Der Systemspeicher **130** schließt Computerspeicherdatenträger in der Form von flüchtigem und/oder nicht-flüchtigem Speicher ein, wie zum Beispiel read only memory (ROM) **131** und random access memory (RAM) **132**. Ein Basic-input/output-System **133** (BIOS) enthält die Basisroutinen, die helfen, Informationen zwischen Elementen innerhalb des Computers **110** zu übertragen, zum Beispiel während des Hochfahrens, und ist üblicherweise in dem ROM **131** gespeichert. RAM **132** enthält üblicherweise Daten und/oder Programmmodule, auf die sofort durch die Prozessoreinheit **120** zugegriffen werden kann und/oder auf denen die Prozessoreinheit **120** derzeit arbeitet. Als Beispiel, und nicht Einschränkung, stellt Fig. 1 ein Betriebssystem **134**, Anwendungsprogramme **135**, andere Programmmodule **136** und Programmdaten **137** dar.

[0029] Der Computer **110** kann ebenso andere entfernbare/nicht-entfernbare, flüchtige/nicht-flüchtige Computerspeicherdatenträger einschließen. Nur als Beispiel stellt Fig. 1 ein Festplattenlaufwerk **140**, das von oder zu nicht-entfernbaren, nicht-flüchtigen magnetischen Datenträgern liest oder schreibt, ein magnetisches Disk-Laufwerk **151**, das von oder zu einer entfernbaren, nicht-flüchtigen magnetischen Disk **152** liest oder schreibt, und ein optisches Disk-Laufwerk **155**, das von oder zu einer entfernbaren, nicht-flüchtigen optischen Disk **156**, wie zum Beispiel einer CD-ROM oder einem anderen optischen Datenträger liest oder schreibt, dar. Andere entfernbare/nicht-entfernbare, flüchtige/nicht-flüchtige Computerspeicherdatenträger, die in der exemplarischen Arbeitsumgebung verwendet werden können, schließen ein, sind aber nicht darauf begrenzt, magnetische Bandkassetten, Flash-Speicherkarten, digital-versatile-disks, digitale Videobänder, solid-state-RAM, solid-state-ROM und ähnliches. Das Festplattenlaufwerk **141** ist üblicherweise mit dem Systembus **121** durch eine Schnittstelle für nicht-entfernbaren Speicher, wie zum Beispiel Schnittstelle **140**, verbunden, und das magnetische Disk-Laufwerk **151** und optische Disk-Laufwerk **155** sind üblicherweise mit dem Systembus **121** durch eine Schnittstelle für entfernbaren Speicher, wie zum Beispiel Schnittstelle **150**, verbunden.

[0030] Die Laufwerke und ihre zugehörigen Computerspeicherdatenträger, die oberhalb diskutiert und in Fig. 1 dargestellt sind, bieten Speicher für computerlesbare Instruktionen, Datenstrukturen, Programmmodule und andere Daten für den Computer **110**. In Fig. 1 zum Beispiel ist das Festplattenlaufwerk **140** so dargestellt, dass das Betriebssystem **144**, Anwendungsprogramme **145**, andere Programmmodule **146** und Programmdaten **147** speichert. Es wird angemerkt, dass diese Komponenten entweder die selben oder verschieden von dem Betriebssystem **134**, Anwendungsprogrammen **135**, anderen Programmmodulen **136** und Programmdaten **137** sein können.

Dem Betriebssystem **144**, Anwendungsprogrammen **145**, anderen Programmmodulen **146** und Programmdaten **147** sind hier unterschiedliche Nummern gegeben worden, um darzustellen, dass sie mindestens unterschiedliche Kopien sind. Ein Benutzer kann Befehle und Informationen in den Computer **20** durch Eingabegeräte, wie zum Beispiel eine Tastatur **162** und Zeigergerät **161**, allgemein als Mause, Trackball oder Touchpad bezeichnet, eingeben. Andere Eingabegeräte (nicht gezeigt) können ein Mikrophon, Joystick, Gamepad, Satellitenschüssel, Scanner oder ähnliches einschließen. Diese und andere Eingabegeräte sind oft mit der Prozessoreinheit **120** durch eine Benutzereingabeschnittstelle **160** verbunden, die mit dem Systembus gekoppelt ist, können aber durch andere Schnittstellen oder Busstrukturen, wie zum Beispiel einem Parallelanschluss, Gameport oder einem Universal Serial Bus (USB) verbunden sein. Ein Monitor **191** oder anderer Typ von Anzeigegerät kann ebenso mit dem Systembus **121** über eine Schnittstelle, wie zum Beispiel einer Videoschnittstelle **190**, verbunden sein. Zusätzlich zu dem Monitor können Computer ebenso andere periphere Ausgabegeräte einschließen, wie zum Beispiel Lautsprecher **197** und Drucker **196**, welche durch eine Ausgabeperipherieschnittstelle **190** verbunden sind.

[0031] Der Computer **110** kann in einer Netzwerkumgebung unter Verwendung logischer Verbindungen zu einem oder mehreren Remote-Computern, wie zum Beispiel einem Remote-Computer **180**, arbeiten. Der Remote-Computer **180** kann ein Personal-Computer, Server, Router, Netzwerk-PC, ein Peer-Gerät oder anderer bekannter Netwerkknoten sein, und schließt üblicherweise viele oder alle der oben mit Bezug auf Computer **110** beschriebenen Elemente ein, obwohl nur ein Datenspeichergerät **181** in Fig. 1 dargestellt worden ist. Die logischen Verbindungen, die in Fig. 1 gezeigt sind, schließen ein Local-Area-Network (LAN) **171** und ein Wide-Area-Network (WAN) **173** ein, können aber auch andere Netzwerke einschließen. Solche Netzwerkumgebungen sind alltäglich in Büros, unternehmensweiten Computernetzwerken, Intranets und dem Internet.

[0032] Wenn er in einer LAN-Netzwerkumgebung verwendet wird, ist der Computer **110** mit dem LAN **171** durch eine Netzschnittstelle oder -adapter **170** verbunden. Wenn er in einer WAN-Netzwerkumgebung verwendet wird, schließt der Computer **110** üblicherweise ein Modem **172** oder andere Mittel zum Herstellen von Datenübertragungen über das WAN **173**, wie zum Beispiel dem Internet, ein. Das Modem **172**, welches intern oder extern sein kann, kann mit dem Systembus **121** über die Benutzereingabeschnittstelle **160** oder einen anderen passenden Mechanismus verbunden sein. In einer Netzwerkumgebung können Programmmodule, die mit Bezug auf Computer **110** gezeigt sind, oder Teile davon, in dem Remote-Datenspeichergerät gespeichert sein. Als Beispiel, und nicht Einschränkung, stellt Fig. 1 Remote-Anwendungsprogramme **185** so dar, dass sie auf dem Speichergerät **181** liegen. Es wird begrüßt, dass die gezeigten Netzwerkverbindungen exemplarisch sind und andere Mittel zum Herstellen einer Datenübertragungsverbindung zwischen den Computern verwendet werden können.

Digitale Signaturunterobjekte

[0033] Wo ein Header-Objekt Unterobjekte und Regionen von Unterobjekten, die, gemäß der Erfindung, geschützt werden müssen, einschließt, kann ein digitales Signaturunterobjekt zu dem Header hinzugefügt werden, um eine Verifizierung zu erlauben, dass die Unterobjekte und Regionen, die signiert sind, nicht verfälscht worden sind. Dieses digitale Signaturunterobjekt kann auf irgendeinem digitalen Signierungsalgorithmus basieren, der als Eingabe irgendwelche Daten nimmt und eine Signatur erzeugt, die später verifiziert werden kann. In einer Ausführungsform ist der verwendete Algorithmus der RSA-Algorithmus. In einer anderen Ausführungsform wird der elliptische Kurvenalgorithmus (elliptic curve algorithm) verwendet. Andere Ausführungsformen können andere Signaturalgorithmen verwenden.

[0034] Bezugnehmend auf Fig. 2 enthält die Datei **200** ein Header-Objekt **210**. Zusätzlich zu den Header-Informationen **215** enthält das Header-Objekt **210** ein Dateieigenschaftsunterobjekt **220**, ein Datenstromeigenschaftenunterobjekt **230**, ein Skriptbefehlsunterobjekt **240** und Inhaltsbeschreibungsunterobjekt **250**. Inhaltsbeschreibungsunterobjekt **250** enthält Informationen über den Titel **252**, Autor **254**, Copyright **256** und Beschreibung **258** des Inhalts. Das Skriptbefehlsunterobjekt **240** enthält eine URL **245**. Die Datei **200** enthält ebenso ein Datenobjekt **290**. Diese Figur ist exemplarisch und es wird erkannt werden, dass andere, als die gezeigten, Kombinationen von Unterobjekten in dem Header-Objekt vorhanden sein können.

[0035] Eine Einheit kann eine Verfälschung von Teilen des Header-Objekts **210** durch das Hinzufügen eines digitalen Signaturunterobjekts **260** verhindern. Das digitale Signaturunterobjekt **260** enthält ein Regionsspezifizierer-Array **264** und eine Signatur **266**. In einer Ausführungsform enthält das digitale Signaturunterobjekt **260** auch Unterzeichnerinformationen **268**. In einer Ausführungsform enthalten die Unterzeichnerinformationen **268** ein oder mehrere Zertifikate, welche verwendet werden können, um die Signatur **266** sicher zu verifizieren.

[0036] Der Prozess zum Erzeugen eines digitalen Signaturunterobjekts ist in Fig. 3 gezeigt. Wie es in Schritt **310** gezeigt ist, entscheidet die Einheit, welche der einen oder mehreren Regionen der Hea-

der-Unterobjekte es signieren wird, und ermittelt die Regionsspezifizierer für diese Regionen. Zum Beispiel, mit Bezug auf Fig. 2, können die zu signierenden Regionen das Skriptbefehlsunterobjekt **230** und die Titel-, Autor- und Copyright-Abschnitte des Inhaltbeschreibungsunterobjekts **250** einschließen. Wieder Bezug nehmend auf Fig. 3 wird in Schritt **320** das Regionsspezifizierer-Array **264** (aus Fig. 2) erzeugt. In Schritt **330** werden die Regionen, die in dem Regionsspezifizierer-Array **264** spezifiziert sind, zusammen mit dem Regionsspezifizierer-Array **264** verkettet (in der Reihenfolge, in der sie in dem Regionsspezifizierer-Array **264** spezifiziert sind). Diese Region wird anschließend signiert **340**, um die Signatur **266** (aus Fig. 2) zu erzeugen.

[0037] Wenn eine Datei, die ein Header-Objekt einschließlich einem digitalen Signaturunterobjekt enthält, verändert wird, kann die Reihenfolge der Unterobjekte verändert werden und zusätzliche Unterobjekte können eingefügt werden. Wenn zusätzliche Regionen oder Unterobjekte verifiziert werden müssen, kann ein neues digitales Signaturunterobjekt hinzugefügt werden.

[0038] Mit Bezug auf Fig. 2 werden, um die Verifizierung des Header-Objekts **210** zu überprüfen, das digitale Signaturunterobjekt **260** und die Regionen, die in dem Regionsspezifizierer-Array **264** spezifiziert sind, verwendet. Wie es in Fig. 4 gezeigt ist, Schritt **410**, werden die Header-Unterobjektsregionen, die in dem Regionsspezifizierer-Array **264** (aus Fig. 2) spezifiziert sind, identifiziert. In Schritt **42C** werden diese Regionen zusammen mit dem Regionsspezifizierer-Array **264** verkettet (in der Reihenfolge, in der sie in dem Regionsspezifizierer-Array **264** spezifiziert sind). In Schritt **430** wird die Signatur **266** (aus Fig. 2) überprüft, um zu ermitteln, ob es eine gültige Signatur für die Verkettung ist.

[0039] In einer Ausführungsform der Erfindung können sowohl Regionen mit Unterobjekten und vollständige Unterobjekte unter Verwendung des digitalen Signaturunterobjekts signiert werden. In einer anderen Ausführungsform können nur vollständige Unterobjekte signiert werden. In einer Ausführungsform der Erfindung kann mehr als eine Region von einem einzelnen Unterobjekt in einem digitalen Signaturunterobjekt signiert werden. In einer Ausführungsform der Erfindung können die Regionen von einem Unterobjekt, das signiert ist, überlappen.

[0040] In einer Ausführungsform der Erfindung muss jedes Header-Objekt mindestens ein digitales Signaturunterobjekt enthalten. Wenn das Header-Objekt kein digitales Signaturunterobjekt enthält, wenn eines erwartet wird, dann kann davon ausgegangen werden, dass das Header-Objekt verfälscht worden ist. Wenn das Header-Objekt ein digitales Signaturunterobjekt enthält, das nicht korrekt verifiziert werden kann oder es nicht von einer vertrauenswürdigen Quelle stammt, kann die Einheit, die die Datei empfängt, die das Header-Objekt enthält, entsprechend verfahren, zum Beispiel, in einer Implementation, durch das Nichtverwenden der Datei. Gemäß dieser Ausführungsform wird eine Überprüfung durchgeführt, um zu sehen, ob irgendein digitales Signaturunterobjekt existiert. Wenn keines existiert, dann schlägt die Verifizierung fehl. Wenn Unterobjekte existieren, wird jedes überprüft, um zu einem Verifizierungsergebnis zu führen.

[0041] In einer Ausführungsform kann irgendeine Datei F, die eine Sammlung von Objekten O_1, O_2, ... O_n ist, gemäß der Erfindung signiert sein. Ein neues Objekt O_{DS} wird erzeugt, welches ein Regionsspezifizierer-Array, das die Objekte oder Regionen von Objekten, die signiert sind, spezifiziert, und eine Signatur für jene Objekte und das Array einschließt.

Exemplarische ASF-Implementation

[0042] In einer Ausführungsform ist die Datei eine ASF-Datei. Die Komponenten eines digitalen Signaturunterobjekts für eine ASF-Datei sind in einer Ausführungsform in Fig. 5 gezeigt. Das digitale Signaturunterobjekt **500** schließt eine GUID **510** ein. Jedes Objekt und Unterobjekt in einer ASF-Datei beginnt mit einer GUID. GUIDs werden verwendet, um alle Objekttypen innerhalb von ASF-Dateien eindeutig zu identifizieren. Jeder ASF-Objekttyp hat seine eigene eindeutige GUID. Jedoch können im Allgemeinen GUIDs nicht verwendet werden, um Unterobjekte innerhalb eines ASF-Headerobjekts eindeutig zu identifizieren, weil mehrere Unterobjekte in einem ASF-Headerobjekt denselben Objekttyp haben können und deshalb die gleiche GUID haben.

[0043] Das nächste Element in dem exemplarischen digitalen Signaturunterobjekt **500** für ASF ist die Unterobjektsgröße **520**. Wieder schließen im Allgemeinen alle ASF-Objekte und Unterobjekte die Größe des Objekts und Unterobjekts ein. Dem Regionsspezifizierer-Array **540**, wie oberhalb beschrieben, geht eine Anzahl von signierten Regionen, die in dem Regionsspezifizierer-Array **530** enthalten sind, voraus. Der Prüfsummen-Algorithmus-Identifizierer **550** und der Signatur-Algorithmus-Identifzierer **560** identifizieren die Prüfsummen- und Signatur-Algorithmen, die in dem digitalen Signaturunterobjekt verwendet wurden. Der Signatur **580** der Regionen und des Regionsspezifizierer-Arrays geht die Länge der Signatur **570** voraus. Die Unterzeichnerinformationen **590** enthalten Informationen, die benötigt werden, um Informationen bezüglich des Unterzeichners zu verifizieren oder zu erhalten. Die Unterzeichnerinformationen **590** können die Identität des Unterzeichners einschließen. In einer Ausführungsform enthält die Unterzeichnerinformation **590** eine Zertifikatskette, die verwendet werden kann, um zu verifizieren,

dass der öffentliche Schlüssel des Unterzeichners von einer vertrauenswürdigen Quelle stammt.

[0044] In der exemplarischen ASF-Implementation enthält jeder Regionsspezifizierer einen Unterobjektsregionsabstand (offset), eine Unterobjektsregionsgröße, eine Prüfsummenlänge und eine Objektprüfsumme. Der Regionsabstand identifiziert, wo die Region in dem Unterobjekt beginnt, und die Regionsgröße identifiziert die Größe der Region. Die Objektsprüfsumme entspricht der Prüfsumme der spezifizierten Region. Dieser Prüfsummen-Algorithmus ist in einer bevorzugten Ausführungsform der Secure-Hash-Algorithm (SHA-1). Dieser Algorithmus ist in der staatlichen Veröffentlichung von Informationsverarbeitungsstandards (Federal Information Processing Standards Publication) 180-1 verfügbar, welche im Internet unter http://www.itl.nist.aov/fipspubs/fip180-1.htm verfügbar ist. In alternativen Ausführungsformen kann irgendein Hash-Algorithmus mit einer geringen Kollisionswahrscheinlichkeit verwendet werden. In einer alternativen Ausführungsform entspricht die Objektprüfsumme der Prüfsumme des Unterobjekts, das die spezifizierte Region enthält.

[0045] Wenn die Signatur überprüft wird, um zu ermitteln, in welchem Unterobjekt die Region liegt (wie in Schritt **410** aus Fig. 4), werden die Header-Unterobjekte untersucht. Für jedes überprüfte Unterobjekt wird eine Prüfsumme gemäß dem in dem Prüfsummen-Algorithmus-Identifizierer **550** spezifizierten Algorithmus berechnet. In der Ausführungsform, wo die Prüfsumme über die Region berechnet wird, wird eine Prüfsumme für die Daten, die in diesem Unterobjekt enthalten sind, welches an dem gegebenen Unterobjekt-Regionsabstand beginnt und sich bis zu der gegebenen Unterobjektregionsgröße erstreckt, berechnet. In der Ausführungsform, wo die Prüfsumme über das gesamte Unterobjekt berechnet wird, wird eine Prüfsumme für das Unterobjekt berechnet. Wenn eine Prüfsumme berechnet wurde, die mit der Prüfsumme in dem Regionsspezifizierer übereinstimmt, ist das richtige Unterobjekt für den Regionsspezifizierer identifiziert worden. Wenn entsprechend jedem Regionsspezifizierer ein Unterobjekt identifiziert worden ist, kann die Signatur überprüft werden.

[0046] In dieser Implementation wird, um ein gesamtes zu signierendes Unterobjekt zu spezifizieren, der Abstand in dem Regionsspezifizierer Null sein, und die Regionsgröße wird gleich der Länge des Unterobjekts sein. In einer anderen Ausführungsform wird die Prüfsumme eher für das gesamte Unterobjekt sein, als für die spezifizierte Region.

[0047] In dieser Ausführungsform kann mehr als ein digitales Signaturunterobjekt in einem Objekt eingeschlossen sein, um die Flexibilität zu erlauben, verschiedene Gebiete von Unterobjekten, die zusammen verifiziert werden, zu haben, und unterschiedliche Einheiten, die Unterobjekte verifizieren, zu haben.

[0048] In anderen Ausführungsformen können andere Verfahren zum Identifizieren der Regionen verwendet werden. In einer Ausführungsform sind Daten, welche das Unterobjekt eindeutig identifizieren können, in dem Regionsspezifizierer zusammen mit Regionsabstands- und -größendaten enthalten.

[0049] In anderen Ausführungsformen können nur gesamte Unterobjekte signiert werden. In einer Ausführungsform schließt der Regionsspezifizierer eine Prüfsumme über das gesamte Unterobjekt ein. In einer anderen Ausführungsform ist ebenso die Länge der Prüfsumme eingeschlossen. In noch einer anderen Ausführungsform werden andere Daten, die das Unterobjekt identifizieren können, in dem Regionsspezifizierer verwendet.

Ergebnis

[0050] Hierin ist ein System und Verfahren zum Datenstrom-Header-Objektschutz. Wie oberhalb erwähnt, können die zugrunde liegenden Konzepte auf irgendwelche Computergeräte oder Systeme angewandt werden, in welchen es wünschenswert ist, Datenstrom-Header-Objekteschutz bereitzustellen, während exemplarische Ausführungsformen der vorliegenden Erfindung in Verbindung mit verschiedenen Computergeräten und Netzwerkarchitekturen beschrieben worden sind. Deshalb können die Techniken zum Bereitstellen von Datenstrom-Header-Objektschutz zugehörig zu der vorliegenden Erfindung auf eine Vielfalt von Anwendungen und Geräten angewandt werden. Zum Beispiel können die Techniken der Erfindung auf das Betriebssystem eines Computergerätes angewandt werden, das als ein getrenntes Objekt auf dem Gerät, als Teil eines anderen Objekts, als ein von einem Server herunterladbares Objekt, als ein "Mittelsmann" zwischen einem Gerät oder Objekt und dem Netzwerk, als ein verteiltes Objekt etc. bereitgestellt ist. Während hier exemplarische Namen und Beispiele repräsentativ für verschiedene Wahlmöglichkeiten gewählt worden sind, sind diese Namen und Beispiele nicht als einschränkend gedacht.

[0051] Die verschiedenen Techniken, die hier beschrieben sind, können in Verbindung mit Hardware oder Software, oder wo angebracht, mit einer Kombination von beidem, implementiert werden. Die Verfahren und die Vorrichtung der vorliegenden Erfindung oder bestimmte Aspekte oder Teile davon können deshalb die Form von Programmcode (das heißt Instruktionen), die in materiellen Datenträgern, wie zum Beispiel Floppy-Disks, CD-ROMs, Festplatten oder anderen maschinenlesbaren Speicherdatenträgern verkörpert sind, annehmen, wobei, wenn der

Programmcode in eine Maschine, wie zum Beispiel einen Computer, geladen und ausgeführt wird, die Maschine eine Vorrichtung zum Praktizieren der Erfindung wird. In dem Fall von Programmcode-Ausführung auf programmierbaren Computern wird das Computergerät im Allgemeinen einen Prozessor, Speicherdatenträger, der durch den Prozessor lesbar ist, (einschließlich flüchtigem und nicht-flüchtigem Speicher und/oder Speicherelementen), mindestens ein Eingabegerät und mindestens ein Ausgabegerät einschließen. Ein oder mehrere Programme, die die Techniken der vorliegenden Erfindung nutzen können, zum Beispiel durch die Verwendung einer Datenverarbeitungs-API oder ähnlichem, werden vorzugsweise in einer Highlevel-Ablauf- oder objektorientierten Programmiersprache implementiert werden, um mit einem Computersystem zu kommunizieren. Jedoch kann/können das/die Programm(e) in Assembly- oder Maschinensprache, wenn gewünscht, implementiert werden. In jedem Fall kann die Sprache eine kompilierte oder interpretierte Sprache sein und mit Hardware-Implementationen kombiniert werden.

[0052] Die Verfahren und die Vorrichtung der vorliegenden Erfindung können ebenso über Datenübertragungen praktiziert werden, die in Form von Programmcode verkörpert werden, der über einige Übertragungsdatenträger übermittelt wird, wie zum Beispiel über elektrische Verkabelung oder Verdrahtung, durch Glasfaserkabel oder über irgendeine andere Form der Übertragung, wobei, wenn der Programmcode empfangen und in eine Maschine geladen und durch die Maschine ausgeführt wird, wie zum Beispiel ein EPROM, ein Gate-Array, ein programmierbares logisches Gerät (programmable logic device-PLD), ein Client-Computer, ein Videorekorder oder ähnliches oder eine empfangende Maschine, die die Signalverarbeitungsmöglichkeiten, wie in exemplarischen Ausführungsformen oberhalb beschrieben, aufweist, eine Vorrichtung zur Praktizierung der Erfindung wird. Wenn er auf einem Allzweckprozessor implementiert ist, vereinigt sich der Programmcode mit dem Prozessor, um eine einzigartige Vorrichtung bereitzustellen, die arbeitet, um die Funktionalität der vorliegenden Erfindung aufzurufen. Zusätzlich können jegliche Speichertechniken, die in Verbindung mit der vorliegenden Erfindung verwendet werden, unveränderlich eine Kombination von Hardware und Software sein.

[0053] Während die vorliegende Erfindung in Verbindung mit den bevorzugten Ausführungsformen der verschiedenen Figuren beschrieben worden ist, gilt es als verstanden, dass, ohne davon abzuweichen, andere ähnliche Ausführungsformen verwendet werden können, oder Modifikationen und Additionen zu der beschriebenen Ausführungsform zum Durchführen der selben Funktion der vorliegenden Erfindung gemacht werden können. Zum Beispiel, während exemplarische Netzwerkumgebungen der Erfindung in dem Kontext einer vernetzten Umgebung beschrieben worden sind, wie zum Beispiel einer Peer-zu-Peer-Netzwerkumgebung, wird ein Fachmann erkennen, dass die Verbindung nicht darauf begrenzt ist, und dass die Verfahren, wie sie in der vorliegenden Anmeldung beschrieben sind, auf irgendein(e) Computergerät oder -umgebung, wie zum Beispiel eine Spielekonsole, tragbaren Computer, portablen Computer, etc., ob verkabelt oder kabellos, angewandt werden können, und auf irgendeine Anzahl von solchen Computergeräten, die über ein Datenübertragungsnetzwerk verbunden sind, und über das Netzwerk interagieren, angewandt werden können. Des Weiteren sollte hervorgehoben werden, dass eine Vielzahl von Computerplattformen, einschließlich Betriebssystemen von tragbaren Geräten und anderen Anwendungsspezifischen Betriebssystemen in Betracht gezogen werden, speziell nachdem die Anzahl der kabellosen Netzwerkgeräte sich kontinuierlich vermehrt. Noch des Weiteren kann die vorliegende Erfindung in oder über eine Vielzahl von Prozessorchips oder -geräten implementiert werden und Speicherung kann ähnlich über eine Vielzahl von Geräten bewerkstelligt werden. Deshalb sollte die vorliegende Erfindung nicht auf eine einzelne Ausführungsform begrenzt sein, aber sollte eher in Breite und Umfang gemäß den anhängigen Ansprüchen ausgelegt werden.

Patentansprüche

1. Verfahren zum Erlauben von Datenverifizierung, wobei ein digitales Objekt (**210**) einer Datenstromdatei (**200**) mindestens ein Unterobjekt (**220**–**260**) umfasst, wobei das Verfahren eine digitale Signatur für mindestens eine Region bereitstellt, wobei jede der mindestens einen Region aus allem oder einem Teil (**252**–**258**) von einem des mindestens einen Unterobjekts besteht, und wobei die Unterobjekte innerhalb des Objekts neu arrangiert werden können, ohne die digitale Signatur ungültig zu machen, wobei das Verfahren umfasst:
Erzeugen (**320**) eines Arrays (**264**, **540**), das für jede der mindestens einen Region einen Regionsspezifizierer, der die Region identifiziert, umfasst;
Verketten (**330**) jeder der mindestens einen Region, die in dem Array spezifiziert ist, mit dem Array;
Erstellen (**340**) einer digitalen Signatur (**580**) basierend auf den verketteten Daten, die jede Region und das Array umfassen; und
Hinzufügen, zu dem digitalen Objekt, eines Signaturunterobjekts (**260**, **500**), das das Array und die digitale Signatur umfasst.

2. Verfahren nach Anspruch 1, wobei jede der mindestens einen Region ein Unterobjekt aus dem mindestens einen Unterobjekt umfasst.

3. Verfahren nach Anspruch 1, wobei jeder der

Regionsspezifizierer eine Prüfsumme umfasst, die gemäß einem Prüfsummenalgorithmus berechnet ist.

4. Verfahren nach Anspruch 3, wobei die Prüfsumme für die Region berechnet ist.

5. Verfahren nach Anspruch 3, wobei die Prüfsumme für das Unterobjekt, das die Region enthält, berechnet ist.

6. Verfahren nach Anspruch 3, wobei das Signaturunterobjekt einen Prüfsummenalgorithmusidentifizierer (**550**) umfasst, der den verwendeten Prüfsummenalgorithmus identifiziert.

7. Verfahren nach Anspruch 3, wobei jeder der Regionsspezifizierer eine Prüfsummenlänge umfasst.

8. Verfahren nach Anspruch 1, wobei das Signaturunterobjekt einen Signaturalgorithmusidentifizierer (**560**) umfasst, der einen Signaturalgorithmus identifiziert, der zum Erstellen einer digitalen Signatur verwendet wird.

9. Verfahren nach Anspruch 1, wobei das Signaturunterobjekt einen Unterzeichneridentifizierer (**268**, **590**) umfasst, der einen Unterzeichner zur Verifizierung der digitalen Signatur identifiziert.

10. Verfahren nach Anspruch 9, wobei der Unterzeichneridentifizierer digitale Zertifikate zum sicheren Identifizieren und Verifizieren des öffentlichen Schlüssels des Unterzeichners umfasst.

11. Verfahren nach Anspruch 1, wobei jeder der Regionsspezifizierer einen Regionsabstand (Offset) umfasst, der die Startstelle der entsprechenden Region in einem Unterobjekt identifiziert.

12. Verfahren nach Anspruch 1, wobei jeder der Regionsspezifizierer eine Regionsgröße umfasst, die die Größe der entsprechenden Region in einem Unterobjekt identifiziert.

13. Verfahren nach Anspruch 1, wobei das Objekt ein Header-Objekt für eine Advanced-Streaming-Format-Datei ist.

14. Verfahren nach Anspruch 13, wobei das neue Objekt des Weiteren einen globalen eindeutigen Identifizierer (**510**) umfasst.

15. Verfahren zum Validieren von Daten, wobei ein digitales Objekt (**210**) einer Datenstromdatei (**220**) mindestens ein Unterobjekt (**220**–**260**) umfasst, wobei das Verfahren eine digitale Signatur (**580**) für mindestens eine Region validiert, wobei jede der mindestens einen Region aus allem oder einem Teil (**252**–**258**) von einem des mindestens einen Unterobjekts besteht, wobei ein Array Regionsspezifizierer für jede der mindestens einen Region umfasst, wobei das Verfahren umfasst:
Identifizieren (**410**) einer Region entsprechend jedem der Regionsspezifizierer;
Erzeugen (**420**) eines Datenobjekts, das das Array und für jeden der Regionsspezifizierer die Region entsprechend des Regionsspezifizierer umfasst; und
Validieren (**430**) der digitalen Signatur, die auf das Datenobjekt angewandt wird.

16. Verfahren nach Anspruch 15, wobei das Objekt ein Header-Objekt für eine Advanced-Streaming-Format-Datei ist.

17. Verfahren nach Anspruch 15, das umfasst:
Ermitteln der Anzahl von digitalen Signaturen, die in dem digitalen Objekt vorhanden sind;
Validieren jeder der digitalen Signaturen.

18. Verfahren nach Anspruch 17, das des Weiteren umfasst:
Zurückgeben eines Fehlerwertes, wenn die Anzahl der digitalen Signaturen, die in dem digitalen Objekt vorhanden sind, null ist.

19. Vorrichtung, die ein digitales Objekt (**210**) einer Datenstromdatei (**200**) einschließt, wobei das digitale Objekt mindestens ein Unterobjekt (**220**–**260**) umfasst, wobei das System eine digitale Signatur für mindestens eine Region bereitstellt, wobei jede der mindestens einen Region aus allem oder einem Teil (**252**–**258**) von einem des mindestens einen Unterobjekts besteht, und wobei die Unterobjekte innerhalb des Objekts neu angeordnet werden können, ohne die digitale Signatur ungültig zu machen, wobei das System umfasst:
Array-Erzeugungs-Mittel zum Erzeugen (**320**) eines Arrays (**264**), das für jede der mindestens einen Region einen Regionsspezifizierer umfasst, der die Region identifiziert;
Verkettungsmittel zum Verketten (**330**) jeder der mindestens einen Region, die in dem Array spezifiziert ist, mit dem Array;
Signierungsmittel zum Erstellen (**340**) einer digitalen Signatur (**580**) basierend auf den verketteten Daten, die jede Region und das Array umfassen; und
Signaturunterobjekts-Hinzufügungsmittel zum Hinzufügen, zu dem digitalen Objekt, eines Signaturunterobjekts (**260**, **500**), das das Array und die digitale Signatur umfasst.

20. Vorrichtung nach Anspruch 19, wobei jede der mindestens einen Region ein Unterobjekt aus dem mindestens einen Unterobjekt umfasst.

21. Vorrichtung nach Anspruch 19, wobei jeder der Regionsspezifizierer eine Prüfsumme umfasst, die gemäß einem Prüfsummenalgorithmus berechnet ist.

22. Vorrichtung nach Anspruch 21, wobei die Prüfsumme für die Region berechnet ist.

23. Vorrichtung nach Anspruch 21, wobei die Prüfsumme für das Unterobjekt, das die Region enthält, berechnet ist.

24. Vorrichtung nach Anspruch 21, wobei das Signaturunterobjekt einen Prüfsummenalgorithmusidentifizierer (**550**) umfasst, der den verwendeten Prüfsummenalgorithmus identifiziert.

25. Vorrichtung nach Anspruch 21, wobei jeder der Regionsspezifizierer eine Prüfsummenlänge umfasst.

26. Vorrichtung nach Anspruch 19, wobei das Signaturunterobjekt einen Signaturalgorithmusidentifizierer (**560**) umfasst, der einen Signaturalgorithmus identifiziert, der zum Erstellen einer digitalen Signatur verwendet wird.

27. Vorrichtung nach Anspruch 19, wobei das Signaturunterobjekt einen Unterzeichnetidentifizierer (**268**, **590**) umfasst, der einen Unterzeichner zur Verifizierung der digitalen Signatur identifiziert.

28. Vorrichtung nach Anspruch 27, wobei der Unterzeichneridentifizierer digitale Zertifikate zum sicheren Identifizieren und Verifizieren des öffentlichen Schlüssels des Unterzeichners umfasst.

29. Vorrichtung nach Anspruch 19, wobei jeder der Regionsspezifizierer einen Regionsabstand (Offset) umfasst, der eine Startstelle der entsprechenden Region in einem Unterobjekt identifiziert.

30. Vorrichtung nach Anspruch 19, wobei jeder der Regionsspezifizierer eine Regionsgröße umfasst, die die Größe der entsprechenden Region in einem Unterobjekt identifiziert.

31. Vorrichtung nach Anspruch 19, wobei das Objekt ein Header-Objekt für eine Advanced-Streaming-Format-Datei ist.

32. Vorrichtung nach Anspruch 31, wobei das neue Objekt des Weiteren einen globalen eindeutigen Identifizierer (**510**) umfasst.

33. Vorrichtung, die ein digitales Objekt (**210**) einer Datenstromdatei (**200**) einschließt, wobei das digitale Objekt mindestens ein Unterobjekt (**220**–**260**) umfasst, wobei das System eine digitale Signatur (**580**) für mindestens eine Region validiert, wobei jede der mindestens einen Region aus allem oder einem Teil (**252**–**258**) von einem des mindestens einen Unterobjekts besteht, wobei ein Array Regionsspezifizierer für jede der mindestens einen Region umfasst, wobei die Vorrichtung umfasst:
Regionsidentifizierungsmittel zum Identifizieren (**410**) einer Region entsprechend jedem der Regionsspezifizierer;
Datenobjekterzeugungsmittel zum Erzeugen (**420**) eines Datenobjekts, das das Array und für jeden der Regionsspezifizierer die Region entsprechend des Regionsspezifizierers umfasst; und
Validierungsmittel zum Validieren (**430**) der digitalen Signatur, die auf das Datenobjekt angewandt wird.

34. Vorrichtung nach Anspruch 33, wobei das Objekt ein Header-Objekt einer Advanced-Streaming-Format-Datei ist.

35. Vorrichtung nach Anspruch 33, umfassend:
Zählungsmittel zum Ermitteln der Anzahl von digitalen Signaturen, die in dem digitalen Objekt vorhanden sind;
Validierungsmittel zum Validieren jeder der digitalen Signaturen.

36. Vorrichtung nach Anspruch 35, die des Weiteren umfasst:
Fehlerzurückgabemittel, das einen Fehlerwert zurückgibt, wenn die Anzahl von digitalen Signaturen, die in dem digitalen Objekt vorhanden sind, null ist.

37. Computerlesbarer Datenträger zum Erlauben von Datenverifizierung, wobei ein digitales Objekt (**210**) einer Datenstromdatei (**200**) mindestens ein Unterobjekt (**220**–**260**) umfasst, wobei der computerlesbare Datenträger eine digitale Signatur für mindestens eine Region bereitstellt, wobei jede der mindestens einen Region aus allem oder einem Teil (**252**–**258**) von einem des mindestens einen Unterobjekts besteht, und wobei die Unterobjekte innerhalb des Objekts neu angeordnet werden können, ohne die digitale Signatur ungültig zu machen, wobei der computerlesbare Datenträger Instruktionen zum Ausführen von Vorgängen aufweist, die umfassen:
Erzeugen (**320**) eines Arrays (**264**), das für jede der mindestens einen Region einen Regionsspezifizierer umfasst, der die Region identifiziert;
Verketten (**330**) jeder der mindestens einen Region, die in dem Array spezifiziert ist, mit dem Array;
Erstellen (**340**) einer digitalen Signatur (**580**) basierend auf den verketteten Daten, die jede Region und das Array umfassen; und
Hinzufügen, zu dem digitalen Objekt, eines Signaturunterobjekts (**260**, **500**), das das Array und die digitale Signatur umfasst.

38. Computerlesbarer Datenträger nach Anspruch 37, wobei jede der mindestens einen Region ein Unterobjekt aus dem mindestens einen Unterobjekt umfasst.

39. Computerlesbarer Datenträger nach Anspruch 37, wobei jeder der Regionsspezifizierer eine Prüfsumme umfasst, die gemäß einem Prüfsum-

menalgorithmus berechnet ist.

40. Computerlesbarer Datenträger nach Anspruch 39, wobei die Prüfsumme für die Region berechnet ist.

41. Computerlesbarer Datenträger nach Anspruch 39, wobei die Prüfsumme für das Unterobjekt, das die Region enthält, berechnet ist.

42. Computerlesbarer Datenträger nach Anspruch 39, wobei das Signaturunterobjekt einen Prüfsummenalgorithmusidentifizierer (**550**) umfasst, der den verwendeten Prüfsummenalgorithmus identifiziert.

43. Computerlesbarer Datenträger nach Anspruch 39, wobei jeder der Regionsspezifizierer eine Prüfsummenlänge umfasst.

44. Computerlesbarer Datenträger nach Anspruch 37, wobei das Signaturunterobjekt einen Signaturalgorithmusidentifizierer (**560**) umfasst, der einen Signaturalgorithmus identifiziert, der zum Erstellen einer digitalen Signatur verwendet wird.

45. Computerlesbarer Datenträger nach Anspruch 37, wobei das Signaturunterobjekt einen Unterzeichneridentifizierer (**268**, **590**) umfasst, der einen Unterzeichner zur Verifizierung der digitalen Signatur identifiziert.

46. Computerlesbarer Datenträger nach Anspruch 45, wobei der Unterzeichneridentifizierer digitale Zertifikate zum sicheren Identifizieren und verifizieren des öffentlichen Schlüssels des Unterzeichners umfasst.

47. Computerlesbarer Datenträger nach Anspruch 37, wobei jeder der Regionsspezifizierer einen Regionsabstand (Offset) umfasst, der die Startstelle der entsprechenden Region in einem Unterobjekt identifiziert.

48. Computerlesbarer Datenträger nach Anspruch 37, wobei jeder der Regionsspezifizierer eine Regionsgröße umfasst, die die Größe der entsprechenden Region in einem Unterobjekt identifiziert.

49. Computerlesbarer Datenträger nach Anspruch 37, wobei das Objekt ein Header-Objekt für eine Advanced-Streaming-Format-Datei ist.

50. Computerlesbarer Datenträger nach Anspruch 49, wobei das neue Objekt des Weiteren einen globalen eindeutigen Identifizierer (**510**) umfasst.

51. Computerlesbarer Datenträger zum Validieren von Daten, wobei ein digitales Objekt (**210**) einer Datenstromdatei (**200**) mindestens ein Unterobjekt (**220**–**260**) umfasst, wobei der computerlesbare Datenträger eine digitale Signatur (**580**) für mindestens eine Region validiert, wobei jede der mindestens einen Region aus allem oder einem Teil (**252**–**258**) von einem des mindestens einen Unterobjekts besteht, wobei ein Array Regionsspezifizierer für jede der mindestens einen Region umfasst, und wobei der computerlesbare Datenträger Instruktionen zum Ausführen von Vorgängen aufweist, die umfassen:
Identifizieren (**410**) einer Region entsprechend jedem der Regionsspezifizierer;
Erzeugen (**420**) eines Datenobjekts, das das Array und für jeden der Regionsspezifizierer die Region entsprechend des Regionsspezifizierers umfasst, und
Validieren (**430**) der digitalen Signatur, die auf das Datenobjekt angewandt wird.

52. Computerlesbarer Datenträger nach Anspruch 51, wobei das Objekt ein Header-Objekt für eine Advanced-Streaming-Format-Datei ist.

53. Computerlesbarer Datenträger nach Anspruch 51, der Instruktionen zum Ausführen von Vorgängen umfasst, die umfassen:
Ermitteln einer Anzahl von digitalen Signaturen, die in dem digitalen Objekt vorhanden sind;
Validieren jeder der digitalen Signaturen.

54. Computerlesbarer Datenträger nach Anspruch 53, wobei der computerlesbare Datenträger Instruktionen zum Ausführen von Vorgängen aufweist, die des Weiteren umfassen:
Zurückgeben eines Fehlerwertes, wenn die Anzahl der digitalen Signaturen, die in dem digitalen Objekt vorhanden sind, null ist.

55. Speicher zum Speichern von Daten zum Zugreifen durch ein Anwendungsprogramm, die eine Datenstruktur (**500**) umfassen, die in dem Speicher gespeichert ist, wobei die Datenstruktur angepasst ist zum Speichern von Verifizierungsinformationen für ein Objekt (**210**) einer Datenstromdatei (**200**), wobei das Objekt mindestens ein Unterobjekt (**220**–**260**) umfasst, während Veränderungen in der Reihenfolge der Unterobjekte erlaubt werden, umfassend:
ein Regionsspezifizierer-Array (**264**, **540**) das mindestens einen Regionsspezifizierer umfasst, wobei jeder Regionsspezifizierer eine Region spezifiziert, die alles oder einen Teil (**252**–**258**) von einem der Unterobjekte umfasst; und
eine digitale Signatur (**580**), die basierend auf verketteten Daten erzeugt wurde, die jede der Regionen und das Regionsspezifizierer-Array umfassen, wobei das Erzeugen der verketteten Daten das Verketten (**330**) jeder der spezifizierten Regionen mit dem Regionsspezifizierer-Array umfasst.

56. Speicher nach Anspruch 55, wobei die Datenstruktur des Weiteren eines oder mehrere der fol-

genden umfasst:
einen globalen eindeutigen Identifizierer (globally unique identifier) (**510**) für die Datenstruktur;
die Größe (**520**) der Datenstruktur;
die Anzahl (**530**) der Regionen in dem Regionsspezifizierer-Array;
einen Prüfsummenalgorithmusidentifizierer (**550**);
einen Signaturalgorithmusidentifizierer (**560**), der den Algorithmus identifiziert, der verwendet wird, um die digitale Signatur zu erzeugen;
eine Signaturlänge (**570**) für die digitale Signatur; und
Unterzeichnerinformation (**268**, **590**) zum Verifizieren der digitalen Signatur.

Es folgen 5 Blatt Zeichnungen

Anhängende Zeichnungen

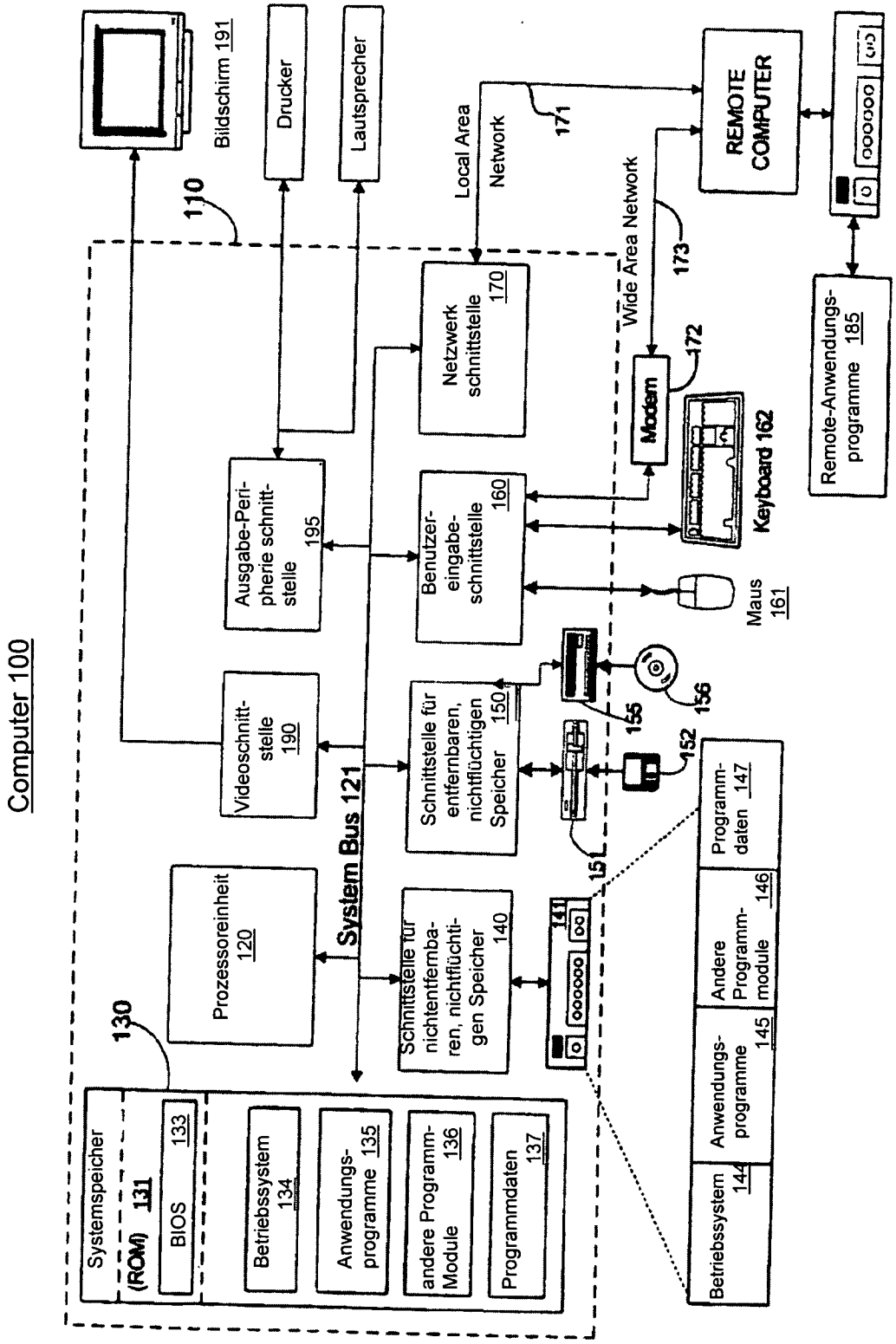

FIG. 1

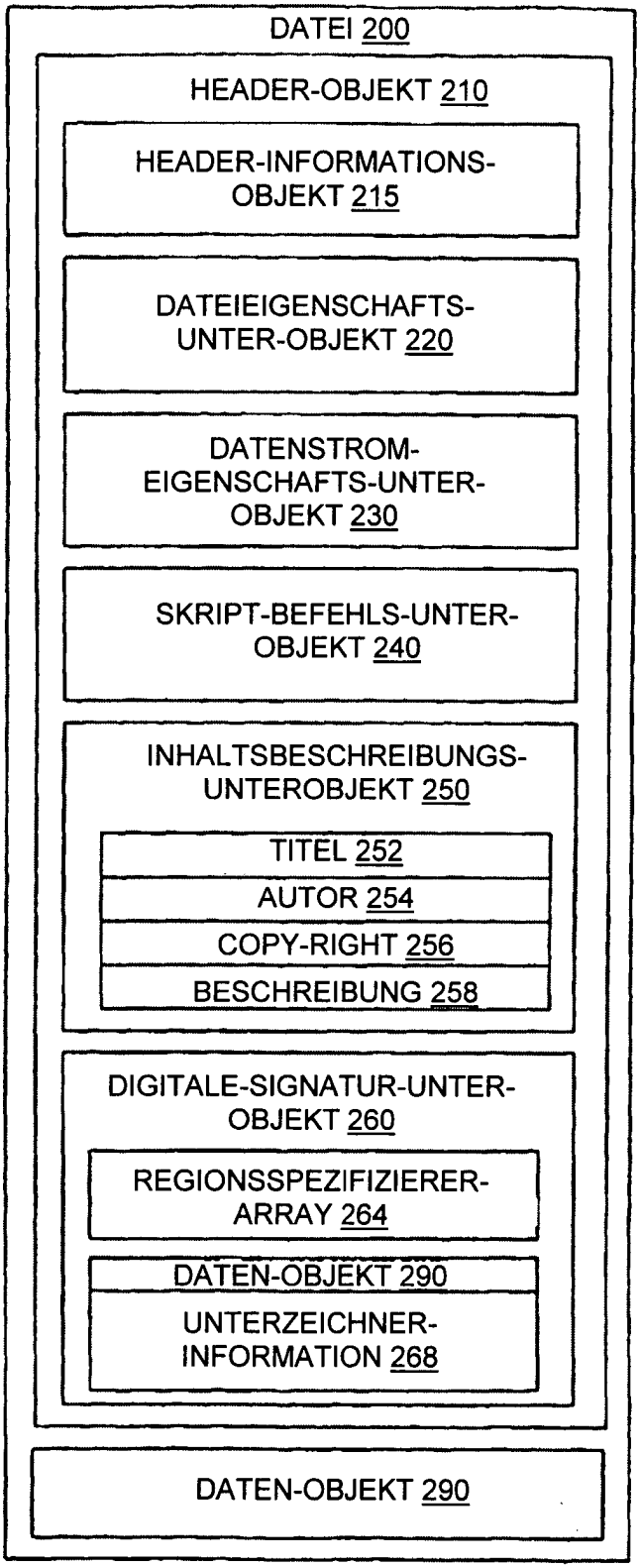
DATEI 200
HEADER-OBJEKT 210
HEADER-INFORMATIONS-OBJEKT 215
DATEIEIGENSCHAFTS-UNTER-OBJEKT 220
DATENSTROM-EIGENSCHAFTS-UNTER-OBJEKT 230
SKRIPT-BEFEHLS-UNTER-OBJEKT 240
INHALTSBESCHREIBUNGS-UNTEROBJEKT 250
TITEL 252
AUTOR 254
COPY-RIGHT 256
BESCHREIBUNG 258
DIGITALE-SIGNATUR-UNTER-OBJEKT 260
REGIONSSPEZIFIZIERER-ARRAY 264
DATEN-OBJEKT 290
UNTERZEICHNER-INFORMATION 268
DATEN-OBJEKT 290

FIG. 2

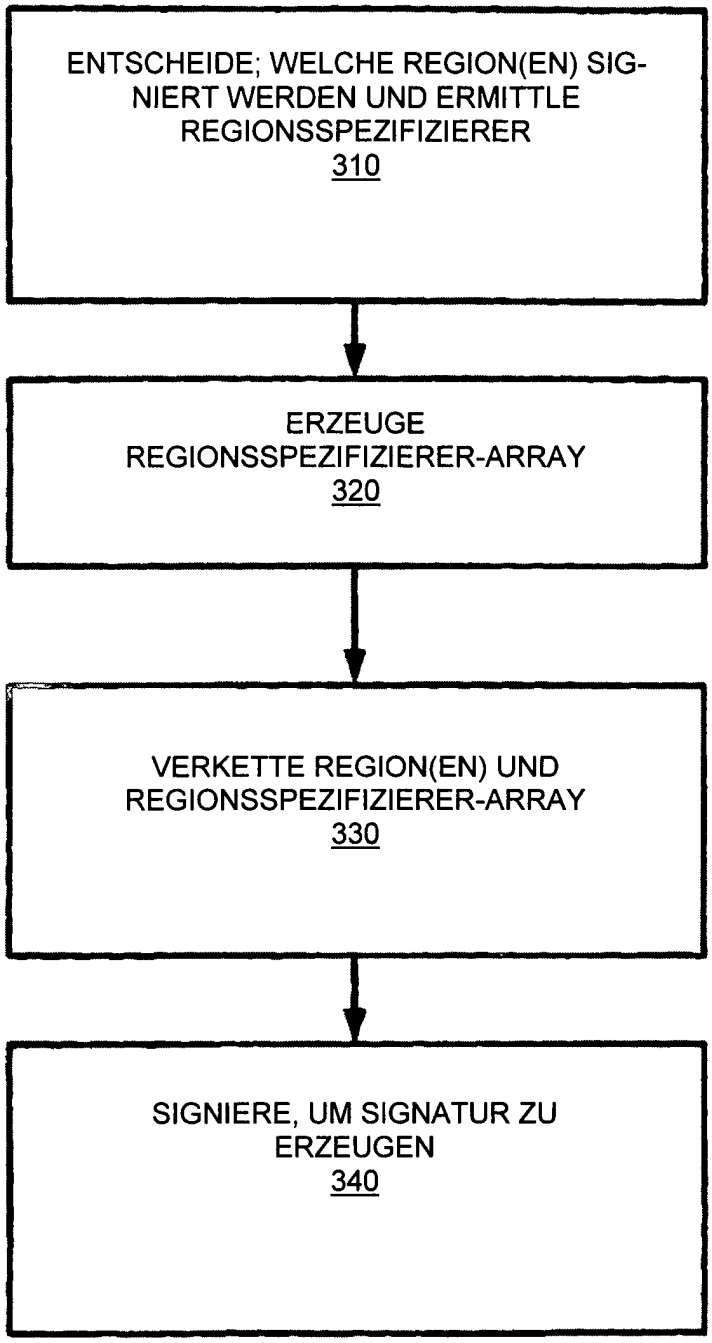
ENTSCHEIDE; WELCHE REGION(EN) SIGNIERT WERDEN UND ERMITTLE REGIONSSPEZIFIZIERER
310
ERZEUGE REGIONSSPEZIFIZIERER-ARRAY
320
VERKETTE REGION(EN) UND REGIONSSPEZIFIZIERER-ARRAY
330
SIGNIERE, UM SIGNATUR ZU ERZEUGEN
340

FIG. 3

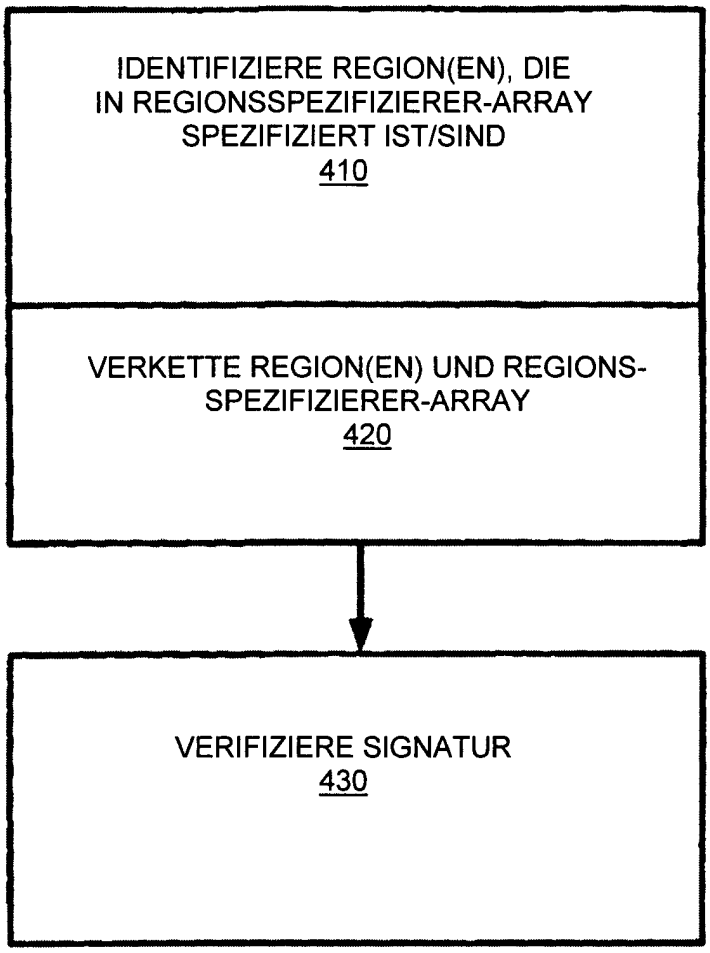
IDENTIFIZIERE REGION(EN), DIE IN REGIONSSPEZIFIZIERER-ARRAY SPEZIFIZIERT IST/SIND
410
VERKETTE REGION(EN) UND REGIONS-SPEZIFIZIERER-ARRAY
420
VERIFIZIERE SIGNATUR
430

FIG. 4

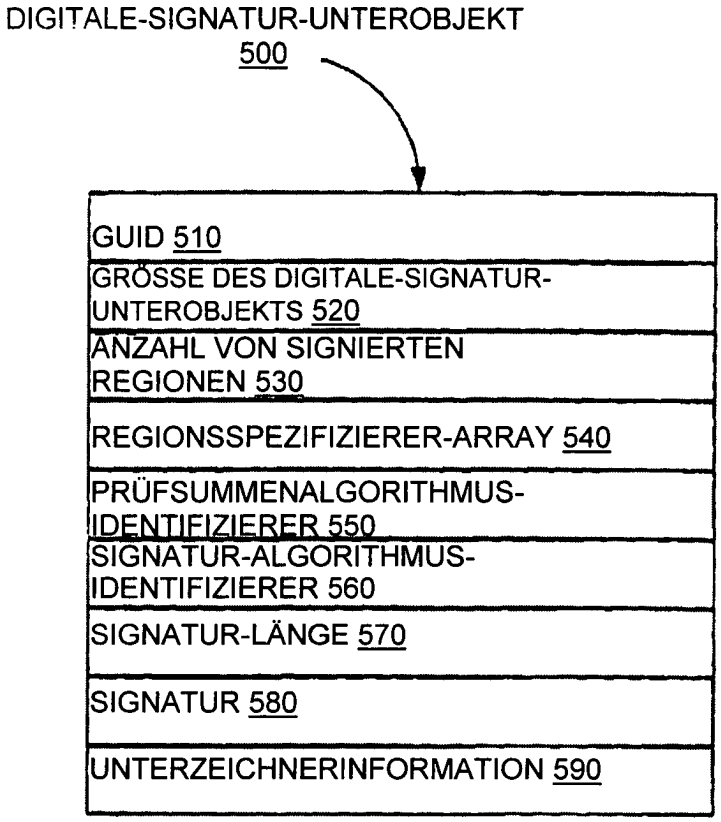
DIGITALE-SIGNATUR-UNTEROBJEKT
500
GUID 510
GRÖSSE DES DIGITALE-SIGNATUR-UNTEROBJEKTS 520
ANZAHL VON SIGNIERTEN REGIONEN 530
REGIONSSPEZIFIZIERER-ARRAY 540
PRÜFSUMMENALGORITHMUS-IDENTIFIZIERER 550
SIGNATUR-ALGORITHMUS-IDENTIFIZIERER 560
SIGNATUR-LÄNGE 570
SIGNATUR 580
UNTERZEICHNERINFORMATION 590

FIG. 5