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

(10) **DE 697 36 138 T2** 2007.04.19

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 251 688 B1**
(21) Deutsches Aktenzeichen: **697 36 138.1**
(96) Europäisches Aktenzeichen: **02 013 520.8**
(96) Europäischer Anmeldetag: **25.04.1997**
(97) Erstveröffentlichung durch das EPA: **23.10.2002**
(97) Veröffentlichungstag
der Patenterteilung beim EPA: **14.06.2006**
(47) Veröffentlichungstag im Patentblatt: **19.04.2007**

(51) Int Cl.[8]: ***H04N 5/00*** (2006.01)
H04N 7/24 (2006.01)

(30) Unionspriorität:
97400650 **21.03.1997** **EP**

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

(73) Patentinhaber:
Thomson Licensing, Boulogne-Billancourt, FR

(74) Vertreter:
Roßmanith, M., Dipl.-Phys. Dr.rer.nat., Pat.-Anw., 30457 Hannover

(72) Erfinder:
Sarfati, Jean-Claude, 93800 Epinay sur Seine, FR;
Meric, Jerome, 60300 Senlis, FR

(54) Bezeichnung: **Datenfernladung**

Beschreibung

[0001] Die vorliegende Erfindung betrifft:

- ein Verfahren zum Herunterladen von Daten auf einen MPEG Empfänger/Decoder,
- einen derartigen MPEG-Empfänger/Decoder für sich und
- ein MPEG-Übertragungssystem.

[0002] Das Aufkommen von digitalen Übertragungssystemen, primär für die Sendung von Fernsehsignalen, insbesondere, jedoch nicht ausschließlich, von Satellitenfernsehsystemen hat die Möglichkeit eröffnet, derartige Systeme für andere Zwecke einzusetzen. Eine dieser besteht darin, eine Interaktivität mit dem Endbenutzer zu bilden.

[0003] Ein Weg, um dies zu tun, besteht darin, eine Anwendung auf dem Empfänger/Decoder zu fahren, durch den die Fernsehsignale empfangen werden. Der Code für die Anwendung könnte ständig in dem Empfänger/Decoder gespeichert sein. Das wäre jedoch ziemlich einschränkend. Vorzugsweise könnte der Empfänger/Decoder in der Lage sein, den Code für eine erforderliche Anwendung herunterzuladen. Auf diese Weise könnte eine größere Vielfalt gebildet werden, und die Anwendungen können, wie gefordert, ohne eine Aktion auf Seiten des Benutzers heruntergeladen werden.

[0004] In einem MPEG-System kann der Anwendungscode in MPEG Tabellen heruntergeladen werden. Es besteht jedoch eine Grenze für die Größe eines Teils des Codes, der durch eine einzige MPEG Tabelle heruntergeladen werden kann. Außerdem wird gefordert, dass eine vollständige Anwendung heruntergeladen wird, bevor sie laufen kann. Das kann eine Verzögerung bewirken, die für den Benutzer inakzeptabel ist. Es besteht daher der Wunsch, in der Lage zu sein, eine Anwendung wie mehrere Module herunterzuladen. Jedoch bildet dieses das Problem, in der Lage zu sein, aus dem MPEG-Bitstrom die Module zu identifizieren und zu extrahieren, die für eine bestimmte Anwendung benötigt werden.

[0005] Die WO 95/33338 lehrt eine dynamische Programmierung eines digitalen Unterhaltungsterminals, die im Bedarfsfall für jeden anderen Service programmiert werden kann, der durch einen oder mehrere Informationsserviceanbieter angeboten wird. Der Steuerprozessor empfängt Benutzereingaben und steuert über einen Zweiwege-Steuerkanal Vorgänge bei dem Terminal sowie die Sendung und den Empfang von Steuersignalen. Der Programmspeicher speichert über einen digitalen Breitbandkanal empfangene Daten als Software, die durch den Steuerprozessor durchführbar ist. Der Steuerprozessor führt die Software aus, die empfangen und in dem Speicher gespeichert wurde zur Steuerung der darauffolgenden Vorgänge des Terminals, einschließlich wenigstens einiger Vorgänge des Audio-/Video-Prozessors und wenigstens einiger Antworten zu den Benutzereingängen.

[0006] Die EP 0 690 400 liefert ein Verfahren zur Erzeugung eines interaktiven Komponentendatenstroms, das ein Anwendungsprogramm für ein Audio-/Video-interaktives (AVI) zusammengesetztes Signal darstellt. Programmdateien, die ein Anwendungsprogramm darstellen, werden erzeugt. Dann werden Flussdaten zur Definition der Datenstruktur der interaktiven Komponente erzeugt. Schließlich wird der Datenstrom durch selektives Einfügen von Programmdateien in den Datenstrom aufgrund der Flussdaten erzeugt. Eine Vorrichtung zur Erzeugung eines interaktiven Komponenten-Datenstroms enthält eine Quelle von Programmdateien auf dem Anwendungsprogramm und eine Quelle von Flussdaten, die die Datenstruktur der interaktiven Komponente definieren. Ein Flussaufbauer (**10**) fügt selektiv Dateien (**30**) von der Quelle von Dateien in die interaktive Komponente aufgrund von Daten von der Flussdatenquelle ein.

[0007] Diese beiden Verfahren bieten keine besondere Sicherheit oder Authentifikation für die heruntergeladenen Anwendungen, was das System verwundbar oder störanfällig macht, insbesondere in einem System mit mehreren Serviceanbietern.

[0008] Die EP 0 752 786 beschreibt ein Verfahren zur Authentifizierung von übertragenen Anwendungen in einem interaktiven Informationssystem. Ein durchführbares interaktives Programm wird aufgeteilt in Module, ähnlich zu Computerdateien, und ein Verzeichnismodul wird gebildet, das die Programmmodule verbindet. Eine Sicherheit für die Durchführungsanwendung wird gebildet durch Anfügung eines signierten oder unterschriebenen Zertifikats auf das Verzeichnismodul, und die Integrität der Module wird überwacht durch Hashing der Module und enthält Hashwerte in dem Verzeichnismodul. Ein Hashwert des Verzeichnismoduls wird erzeugt, verschlüsselt und dem Verzeichnismodul hinzugefügt. Bei einem Empfänger wird das Zertifikat decodiert und für eine Authentifizierung geprüft. Wenn das Zertifikat natürlich (genuine) ist, kann das Programm durchgeführt werden, jedoch nur dann, wenn durch die Empfänger erzeugten Hashwerte identisch zu denjenigen in dem Verzeichnismodul sind.

[0009] Während dieses Verfahren einen ziemlich guten Schutz für die heruntergeladene Software bildet, ist es eine bekannte Tatsache, dass so genannte Hacker häufig versuchen, jedes ihnen verfügbare System zu "hack", das heißt es missbräuchlich zu benutzen. Die Sicherheit dieses Verfahrens kann, nach dem was wir wissen, ausreichend sein, jedoch werden weitere Verbesserungen bei wenigstens einem durch die vorliegende Erfindung gebildeten er-

wünscht.

[0010] Gemäß einem ersten Aspekt der vorliegenden Erfindung ist ein Verfahren vorgesehen zum Herunterladen wenigstens eines Teils einer Anwendung zu einem MPEG Empfänger/Decoder, das folgende Schritte enthält: Aufteilung der Anwendung auf mehrere Module, Formatierung jedes Moduls als eine entsprechende MPEG Tabelle, wobei die Tabellen dieselbe Tabellenidentifikation ("TID") und jeweils verschiedene Tabellenidentifikationserweiterungen ("TID-extensions") oder eine andere vorbestimmte TID-Erweiterung haben, Erzeugung einer Verzeichnis-MPEG Tabelle für die Module mit derselben genannten TID und eine Tabelle für Module mit derselben TID und der vorbestimmten TID-Erweiterung. Das Verzeichnis enthält für jedes der Module einen Namen dieses Moduls und die jeweilige TID-Erweiterung und Mittel zur zyklischen Übertragung der Verzeichnis-MPEG Tabelle und der Modul-MPEG Tabellen in einem MPEG-Bitstrom.

[0011] Vorzugsweise enthält das System außerdem Mittel zur Erzeugung einer Versions-Identifikation für die Verzeichnis-MPEG-Tabelle, wobei die je Verzeichnis-MPEG-Tabelle erzeugenden Mittel in der Verzeichnis-MPEG-Tabelle die erzeugte Versionsidentifikation dafür einfügen.

[0012] Die das Modul formatierende Mittel können wenigstens eine der Modul-MPEG-Tabellen als mehrere MPEG-Abschnitte formatieren, wobei jede in einem vorbestimmten Teil dafür eine Identifikation des MPEG-Abschnitts in der MPEG Tabelle und eine Anzeige der Zahl der MPEG-Abschnitte in dieser MPEG Tabelle enthält.

[0013] In erwünschter Weise sollte der Empfänger/Decoder geschützt sein gegen das Herunterladen von unberechtigten Anwendungen, die zum Beispiel einen Virus enthalten könnten. Daher können die Konzepte der Verschlüsselung und Signierung wenigstens eines Teils des Anwendungscodes in Erwägung gezogen werden.

[0014] Gemäß einem vierten Aspekt der vorliegenden Erfindung ist ein Verfahren zum Herunterladen von Daten zu einem MPEG Empfänger/Decoder vorgesehen, das folgende Schritte enthält: Erzeugung einer Signatur für die herunterzuladenden Daten, Verschlüsselung der Signatur durch einen privaten Schlüssel, Formatierung der herunterzuladenden Daten, der verschlüsselten Signatur und einer Identifikation für den privaten Schlüssel als eine MPEG Tabelle, Übertragung der MPEG Tabelle, und beim Empfänger/Decoder: Empfang der MPEG-Tabelle, Auswahl eines der mehreren öffentlichen Schlüssel entsprechend der Schlüsselidentifikation in der empfangenen MPEG Tabelle, Entschlüsselung der verschlüsselten Signatur in der empfangenen MPEG Tabelle durch den gewählten öffentlichen Schlüssel zur Bildung einer entschlüsselten Signatur, Erzeugung einer Signatur für die Daten in der empfangenen MPEG Tabelle und Vergleich der entschlüsselten Signatur und der Signatur, die beim Empfänger/Decoder für die empfangenen Daten erzeugt wird.

[0015] Daher kann der Empfänger/Decoder dazu dienen, die Anwendungen mit verschlüsselten Signaturen von mehr als einer Quelle herunterzuladen, ohne dass die Quellen die privaten Schlüssel jeweils der anderen kennen müssen.

[0016] Das Verfahren kann außerdem die Schritte zum Herunterladen zu dem Empfänger/Decoder einer Anwendung enthalten, die eine Signatur aufweist, die durch einen privaten Schlüssel mit einer vorbestimmten Schlüsselidentifikation verschlüsselt ist, Ablauf der Anwendung bei dem Empfänger/Decoder, um zu bewirken, dass der Empfänger/Decoder einen weiteren Schlüssel empfängt, Speicherung des weiteren Schlüssels in einem Bereich eines flüchtigen Speichers des Empfänger/Decoders. In diesem Fall kann, während des Schritts des Ablaufs der Anwendung, außerdem der Schlüssel örtlich dem Empfänger/Decoder zugeführt werden, zum Beispiel über einen parallelen Anschluss, einem seriellen Anschluss oder einen Smart Card-Leser des Empfänger/Decoders. Wenn der Empfänger/Decoder einen Modemanschluss aufweist, ist der Empfänger/Decoder vorzugsweise dafür vorgesehen, zu verhindern, dass ein weiterer Schlüssel über das Modem zugeführt wird.

[0017] Diese Merkmale ermöglichen es einem Hersteller, der einen Empfänger/Decoder testen möchte, einen Schlüssel zu dem Empfänger/Decoder herunterzuladen.

[0018] Das Verfahren kann ferner die Schritte beim Empfänger/Decoder des Nachschlagens in einem geschützten Bereich des Speichers des Empfängers/Decoders in einer Validationsmarkierung für den gewählten öffentlichen Schlüssel und die Verhinderung oder Abbrechung des Herunterladens der Daten enthalten, wenn die nachgeschlagene Markierung nicht gesetzt ist.

[0019] Daher kann, wenngleich mehrere öffentliche Schlüssel permanent in dem Speicher des Empfängers/Decoders gebildet werden können, einer von ihnen selektiv gesperrt werden, was zum Beispiel notwendig ist, wo die Privatsphäre für einen bestimmten öffentlichen Schlüssel verletzt wird, oder wo zwei Operatoren, die dieselben Schlüssel benutzt haben, sich für getrennte Schlüssel entscheiden.

[0020] In dem Fall, wo der Empfänger/Decoder vorgesehen ist zum Herunterladen einer Anwendung mit einer Signatur, verschlüsselt mit einem privaten

Schlüssel mit einer vorbestimmten Schlüsselidentifikation, wie oben, in dem geschützten Bereich eines Speichers des Empfängers/Decoders der private Schlüssel mit der vorgegebenen Schlüsselidentifikation eine Validationsmarkierung haben kann, die durch die Anwendung geändert werden kann, und eine Möglichkeit zum Empfang eines derartigen weiteren Schlüssels wird bestimmt in Abhängigkeit von dem Zustand der Validationsmarkierung.

[0021] Diese letzteren Merkmale von Validationsmarkierungen für die öffentlichen Schlüssel können unabhängig von dem vierten Aspekt der Erfindung gebildet werden. Daher bildet ein fünfter Aspekt der vorliegenden Erfindung ein Verfahren zum Herunterladen von Daten zu einem MPEG Empfänger/Decoder mit folgenden Schritten: Formatierung der herunterzuladenden Daten, der verschlüsselten Signatur und einer Identifikation für den privaten Schlüssel als eine MPEG Tabelle, Übertragung der MPEG Tabelle und beim Empfänger/Decoder: Empfang der MPEG Tabelle, Nachschlagen in einem geschützten Bereich des Speichers des Empfängers/Decoders, eine Validationsmarkierung für einen öffentlichen Schlüssel entsprechend dem privaten Schlüssel, identifiziert in der empfangenen MPEG Tabelle, und, wenn die nachgeschlagene Markierung gesetzt ist: – Entschlüsselung der verschlüsselten Signatur in der empfangenen MPEG Tabelle durch den öffentlichen Schlüssel entsprechend dem in der empfangenen MPEG Tabelle identifizierten privaten Schlüssel zur Bildung einer entschlüsselten Signatur, Erzeugung einer Signatur für die Daten MPEG-Tabelle und Vergleich der entschlüsselten Signatur und der beim Empfänger/Decoder erzeugten Signatur für die empfangenen Daten.

[0022] Die Verfahren des vierten oder fünften Aspekts der Erfindung enthalten vorzugsweise außerdem folgende Schritte: – Erzeugung eines Validationscodes für die herunterzuladenden Daten, wobei der Validationscode mit der Signatur in dem Verschlüsselungsschritt verschlüsselt und mit der Signatur in dem Entschlüsselungsschritt entschlüsselt wurde, Nachschlagen eines gespeicherten Validationscodes in einem geschützten Bereich des Empfänger/Decoders und Vergleich des nachgeschlagenen mit dem entschlüsselten Validationscode.

[0023] Daher kann der Empfänger/Decoder dafür eingerichtet werden, nur bestimmte Anwendungen oder Typen von Anwendungen zu empfangen.

[0024] Diese Merkmale können unabhängig von dem vierten und fünften Aspekt der Erfindung vorgesehen sein. Daher bildet ein sechster Aspekt der vorliegenden Erfindung ein Verfahren zum Herunterladen von Daten zu einem MPEG Empfänger/Decoder mit folgenden Schritten: – Erzeugung eines Berechtigungscodes für die herunterzuladenden Daten, Erzeugung einer Signatur für die herunterzuladenden Daten oder einen Teil davon, Verschlüsselung des Berechtigungscodes und der Signatur durch einen privaten Schlüssel, Formatierung der herunterzuladenden Daten und des verschlüsselten Berechtigungscodes und Signatur wenigstens einer MPEG Tabelle, Übertragung für jede MPEG Tabelle und bei dem Empfänger/Decoder: – Empfang der oder jeder MPEG Tabellen, Entschlüsselung des verschlüsselten Berechtigungscodes und Signatur in den empfangenen MPEG Tabellen durch einen öffentlichen Schlüssel entsprechend dem privaten Schlüssel, Nachschlagen eines gespeicherten Berechtigungscodes in einem geschützten Bereich des Speichers des Empfängers/Decoders, Vergleich des nachgeschlagenen Berechtigungscodes und des entschlüsselten Berechtigungscodes, Erzeugung einer Signatur für die Daten in der empfangenen MPEG-Tabelle(n) oder des Teilers davon und Vergleich der entschlüsselten Signatur mit der bei dem Empfänger/Decoder für die empfangenen Daten erzeugten Signatur.

[0025] Vorzugsweise enthält das Verfahren außerdem den Schritt der Verhinderung oder des Abbruchs des Herunterladens der Daten, wenn, in dem Berechtigungscode-Vergleichsschritt, der nachgeschlagene Berechtigungscode und der entschlüsselte Berechtigungscode nicht miteinander übereinstimmten.

[0026] In dem vierten bis sechsten Aspekt der Erfindung kann vorgesehen sein, dass die Signatur der herunterzuladenden Daten in einem Datenblock verschlüsselt wird, der andere Daten enthält, mit einem gewählten Offset zwischen dem Start des Datenblocks und dem Start der Signatur, und der verschlüsselte Datenblock wird in dem Entschlüsselungsschritt bei dem Empfänger/Decoder entschlüsselt, und ferner mit den Schritten, bei dem Empfänger/Decoder, des Nachschlagens wenigstens eines gespeicherten Offset in einem geschützten Bereich des Speichers des Empfänger/Decoders und Extrahierung der Signatur von dem entschlüsselten Datenblock durch einen Nachschlageoffset von dem Start des entschlüsselten Datenblocks.

[0027] Daher kann die Signatur unter den anderen Dummydaten verdeckt sein, was es schwieriger macht, die Lage der Signatur zu erkennen. Alternativ oder zusätzlich ermöglicht dieses Merkmal, dass die Daten nur einer oder mehreren bestimmten Gruppen von Empfängern/Decodern verfügbar gemacht werden.

[0028] Diese Merkmale können unabhängig von dem vierten bis sechsten Aspekt der Erfindung durchgeführt werden. Daher liefert ein siebter Aspekt der vorliegenden Erfindung ein Verfahren zum Herunterladen von Daten in einen MPEG Empfänger/Decoder mit folgenden Schritten: – Erzeugung einer Si-

gnatur für die herunterzuladenden Daten, einschließlich der Signatur und anderer Daten in einem Datenblock mit einem gewählten Offset zwischen dem Start des Datenblocks und dem Start der Signatur, Verschlüsselung des Datenblocks durch einen privaten Schlüssel, Formatierung der herunterzuladenden Daten und des verschlüsselten Datenblocks als eine MPEG Tabelle, Übertragung der MPEG Tabelle und beim Empfänger/Decoder: – Empfang der MPEG Tabelle, Entschlüsselung des verschlüsselten Datenblocks in der empfangenen MPEG Tabelle mit einem öffentlichen Schlüssel entsprechend dem privaten Schlüssel, Nachschlagen wenigstens eines gespeicherten Offsets in einem geschützten Bereich des Speichers des Empfänger/Decoders, Extrahierung der Signatur aus dem entschlüsselten Datenblock durch einen nachgeschlagenen Offset von dem Start des entschlüsselten Datenblocks, Erzeugung einer Signatur für die Daten in der empfangenen MPEG Tabelle und Vergleich der aus dem entschlüsselten Datenblock extrahierten Signatur mit der beim Empfänger/Decoder für die empfangenen Daten erzeugten Signatur.

[0029] In dem Fall, wo der geschützte Bereich des Speichers wenigstens zwei derartige gespeicherte Offsets enthält, wenn in dem Vergleichsschritt die extrahierte Signatur und die erzeugte Signatur nicht übereinstimmten, enthält das Verfahren vorzugsweise die Schritte der Wiederholung des Nachschlagens, Extrahierung und Vergleich der Schritte durch einen anderen gespeicherten Offset.

[0030] Wenigstens einige der anderen Daten in dem Datenblock können so genannte Dummy- oder willkürliche Daten sein, jedoch, wenn das so ist, wiederholt vorzugsweise kein Abschnitt der Dummydaten die Signatur.

[0031] In dem vierten bis siebten Aspekt der Erfindung können die Daten als mehrere Datenmodule heruntergeladen werden, und das Verfahren kann die folgenden Schritte enthalten: – Erzeugung einer Modulsignatur für jedes herunterzuladende Datenmodul, Formatierung der Module als jeweilige Modul-MPEG Tabellen, Erzeugung eines Verzeichnisses mit einer Identifikation für jede Modul-MPEG Tabelle und die jeweilige Signatur, wobei das Verzeichnis Gegenstand der den Schritt erzeugenden Signatur ist, und bei dem Empfänger/Decoder: – Erzeugung einer jeweiligen Modulsignatur für jedes der Module in den empfangenen Modul-MPEG Tabellen und Vergleich jeder Modulsignatur in der empfangenen Verzeichnis-MPEG Tabelle mit der entsprechenden, beim Empfänger/Decoder erzeugten Modulsignatur.

[0032] Daher wird, wenngleich die herunterzuladenden Daten aus mehreren Modulen bestehen, nur ein einziger Verschlüsselungsvorgang benötigt, um die Module zu verschlüsseln, und es wird nur ein einziger Entschlüsselungsvorgang benötigt, um die Signaturen prüfen zu können.

[0033] Diese Merkmale können unabhängig von dem vierten bis siebten Aspekt durchgeführt werden. Daher liefert ein achter Aspekt der vorliegenden Erfindung ein Verfahren zum Herunterladen von mehreren Datenmodulen zu einem MPEG-Empfänger/Decoder mit folgenden Schritten: – Erzeugung einer Modulsignatur für jedes herunterzuladende Datenmodul, Formatierung der Datenmodule als entsprechende Modul-MPEG Tabellen, Erzeugung eines Verzeichnisses mit einer Identifikation für jede Modul-MPEG Tabelle und jeweilige Signatur, Erzeugung einer Verzeichnissignatur für das Verzeichnis, Verschlüsselung der Verzeichnissignatur durch einen privaten Schlüssel, Formatierung der Verzeichnissignatur und der verschlüsselten Signatur als eine Verzeichnis-MPEG Tabelle, Übertragung der Verzeichnis- und Modul-MPEG Tabelle und beim Empfänger/Decoder: – Empfang des Verzeichnisses und der Modul-MPEG Tabellen, Entschlüsselung der verschlüsselten Verzeichnissignatur in der empfangenen Verzeichnis-MPEG Tabelle durch einen dem privaten Schlüssel entsprechenden öffentlichen Schlüssel, Erzeugung einer Verzeichnissignatur für die in der empfangenen Verzeichnis-MPEG Tabelle, Vergleich der entschlüsselten Verzeichnissignatur und der beim Empfänger/Decoder erzeugten Verzeichnissignatur, Erzeugung einer entsprechenden Modulsignatur für jedes der Module in den empfangenen Modul-MPEG Tabellen, und Vergleich jeder Modulsignatur in der empfangenen Verzeichnis-MPEG Tabelle mit der beim Empfänger/Decoder erzeugten jeweiligen Modulsignatur.

[0034] Das Verfahren enthält außerdem die Schritte der Verhinderung oder des Abbruchs des Herunterladens eines derartigen Datenmoduls, wenn in dem Modulsignaturvergleichsschritt, wenn die Modulsignatur in der empfangenen Verzeichnis-MPEG Tabelle und die entsprechende, bei dem Empfänger/Decoder für dieses Modul erzeugte Modulsignatur nicht miteinander übereinstimmen.

[0035] Die oben beschriebenen Verfahren enthalten vorzugsweise den Schritt der Verhinderung oder des Abbruchs des Herunterladens der Daten, wenn in dem Vergleichsschritt(en) die oder jede entschlüsselte Signatur und die erzeugte Signatur nicht miteinander übereinstimmen.

[0036] Gemäß einem neunten Aspekt der vorliegenden Erfindung ist ein MPEG Empfänger/Decoder zur Anwendung in dem durchführenden Teil des Verfahrens des vierten Aspekts der Erfindung vorgesehen, enthaltend: Mittel zum Empfang derartiger MPEG Tabelle, Mittel zur Speicherung von mehreren öffentlichen Schlüsseln und einer Identifikation für jeden der

öffentlichen Schlüssel und Verarbeitungsmittel, die dafür programmiert sind, einen der gespeicherten öffentlichen Schlüssel, entsprechend der Schlüsselidentifikation in der empfangenen MPEG Tabelle, zu wählen, zur Entschlüsselung der verschlüsselten Signatur in der empfangenen MPEG Tabelle durch den gewählten öffentlichen Schlüssel zur Bildung einer entschlüsselten Signatur, zur Erzeugung einer Signatur für die Daten in der empfangenen MPEG Tabelle, und Vergleich der entschlüsselten Signatur und der bei dem Empfänger/Decoder für die empfangenen Daten erzeugten Signatur.

[0037] Die Schlüsselspeicherungsmittel werden vorzugsweise durch ein ROM gebildet, und die Identifikation für jeden der öffentlichen Schlüssel kann durch die Speicherlage dieses öffentlichen Schlüssels in den Schlüsselspeichermitteln geliefert werden.

[0038] Der Empfänger/Decoder kann außerdem einen Bereich eines flüchtigen Speichers enthalten, und die Verarbeitungsmittel können eine Anwendung mit einer Signatur herunterladen, die einen privaten Schlüssel mit einer vorbestimmten Schlüsselidentifikation zum Ablauf der Anwendung besitzt, die bewirkt, dass der Empfänger/Decoder einen weiteren Schlüssel empfängt und der weitere Schlüssel in dem Bereich des flüchtigen Speichers gespeichert wird.

[0039] Der Empfänger/Decoder kann ferner Mittel enthalten zum Empfang eines derartigen weiteren Schlüssels, der lokal dem Empfänger/Decoder zugeführt wird, wie ein paralleler Anschluss, ein serieller Anschluss und/oder Smart Card-Leser des Empfänger/Decoders. Der flüchtige Speicher wird vorzugsweise durch ein RAM gebildet. Wenn der Empfänger/Decoder einen Modemanschluss aufweist, wird wieder der Empfänger/Decoder vorzugsweise so ausgebildet, dass ein weiterer Schlüssel über das Modem zugeführt wird.

[0040] Der Empfänger/Decoder kann außerdem einen geschützten Speicherbereich enthalten zur Speicherung einer Berechtigungsmarkierung für jeden der wenigstens einigen öffentlichen Schlüssel, und die Verarbeitungsmittel können dafür programmiert sein, in dem geschützten Speicherbereich die Berechtigungsmarkierung für einen derartigen gewählten öffentlichen Schlüssel nachzuschlagen und ein Herunterladen der Daten zu verhindern oder abzubrechen, wenn die Nachschlagemarkierung nicht gesetzt ist.

[0041] Der Empfänger/Decoder kann außerdem einen geschützten Bereich des Speichers enthalten für die Speicherung einer Berechtigungsmarkierung für den privaten Schlüssel mit der vorbestimmten Schlüsselidentifikation, und die Verarbeitungsmittel können dazu dienen, wenn diese Anwendung zur Änderung der Gültigkeitsmarkierung in Betrieb ist, zur Freigabe des weiteren Schlüssels, um in Abhängigkeit von dem Zustand dieser Markierung gespeichert zu werden.

[0042] Das letztere Merkmal kann unabhängig von dem neunten Aspekt erfolgen. Daher liefert ein zehnter Aspekt der vorliegenden Erfindung einen MPEG Empfänger/Decoder mit: Mitteln zum Empfang derartiger MPEG Tabellen, Mitteln zur Speicherung eines öffentlichen Schlüssels und einer Identifikation für den öffentlichen Schlüssel und einem geschützten Bereich des Speichers für die Speicherung einer Berechtigungsmarkierung für den öffentlichen Schlüssel, und Verarbeitungsmittel, die dafür programmiert sind, in dem geschützten Bereich des Speichers eine Berechtigungsmarkierung für den öffentlichen vorzusehen, der dem in der empfangenen MPEG Tabelle identifizierten Schlüssel entspricht, und, wenn die Nachschlagemarkierung besetzt ist, zur Entschlüsselung der verschlüsselten Signatur in der empfangenen MPEG Tabelle durch den öffentlichen Schlüssel, entsprechend dem in der empfangenen MPEG Tabelle identifizierten privaten Schlüssel zur Bildung einer entschlüsselten Signatur und zur Erzeugung einer Signatur für die in der empfangenen MPEG-Tabelle empfangenen Daten und zum Vergleich der entschlüsselten Signatur mit der durch den Empfänger/Decoder für die empfangenen Daten erzeugten Signatur.

[0043] Der Speicher zur Speicherung der Schlüsselberechtigungsmarkierung(en) erfolgt vorzugsweise durch einen neubeschreibbaren, nicht-flüchtigen Speicher.

[0044] In dem Fall, wo mehrere derartige öffentliche Schlüssel gespeichert werden, ist der Speicher für die Speicherung der Berechtigungsmarkierung(en) vorzugsweise als eine so genannte Bitmap angeordnet.

[0045] Der Empfänger/Decoder des neunten oder zehnten Aspekts der Erfindung kann außerdem einen geschützten Speicherbereich zur Speicherung eines Berechtigungscodes enthalten, und die Verarbeitungsmittel können programmiert sein zur Entschlüsselung des Berechtigungscodes in einer derartigen empfangenen MPEG Tabelle zum Nachschlagen des gespeicherten Berechtigungscodes und zum Vergleich des nachgeschlagenen Berechtigungscodes mit dem entschlüsselten Berechtigungscode.

[0046] Das letztgenannte Merkmal kann unabhängig von dem neunten oder zehnten Aspekt der Erfindung gebildet werden. Daher bildet ein elfter Aspekt der vorliegenden Erfindung einen MPEG-Empfänger/Decoder mit: Mitteln zum Empfang derartiger MPEG Tabellen, Mitteln zur Speicherung eines öffentlichen Schlüssels und einer Identifikation für den

öffentlichen Schlüssel, einem geschützten Speicherbereich zur Speicherung eines Berechtigungscodes und Verarbeitungsmitteln, die programmiert sind zur Entschlüsselung des verschlüsselten Validationscodes und der Signatur in derartigen empfangenen MPEG Tabellen, unter Benutzung des gespeicherten öffentlichen Schlüssels entsprechend dem privaten Schlüssel, Nachschlagen des gespeicherten Berechtigungscodes in dem geschützten Bereich des Speichers, zum Vergleich des nachgeschlagenen Berechtigungscodes mit dem entschlüsselten Berechtigungscode, zur Erzeugung einer Signatur für die Daten in der empfangenen MPEG Tabelle oder dem Teil davon und zum Vergleich der entschlüsselten Signatur mit der durch den Empfänger/Decoder für die empfangenen Daten erzeugten Signatur.

[0047] Die Verarbeitungsmittel sind vorzugsweise programmiert zur Verhinderung oder zum Abbruch des Herunterladens der Daten, wenn der nachgeschlagene Berechtigungscode und der entschlüsselte Berechtigungscode nicht miteinander übereinstimmen.

[0048] Der Speicher für die Speicherung der Berechtigungscodes wird vorzugsweise durch einen neubeschreibbaren, nicht-flüchtigen Speicher gebildet und kann als so genannte Bitmap ausgebildet sein.

[0049] Der Empfänger/Decoder des neunten bis elften Aspekts der Erfindung kann außerdem einen geschützten Speicherbereich enthalten zur Speicherung wenigstens eines Offsets, und die Verarbeitungsmittel können programmiert sein zur Entschlüsselung des verschlüsselten Datenblocks in einer derartigen empfangenen MPEG Tabelle zum Nachschlagen des in dem geschützten Speicherbereich gespeicherten Offset und zur Extrahierung der Signatur von dem entschlüsselten Datenblock durch den nachgeschlagenen Offset von dem Start des entschlüsselten Datenblocks.

[0050] Dieses letztgenannte Merkmal kann unabhängig von dem neunten bis elften Aspekt der Erfindung gebildet werden. Demgemäß liefert ein zwölfter Aspekt der Erfindung einen MPEG Empfänger/Decoder mit: Mitteln zum Empfang derartiger MPEG Tabellen, Mittel zur Speicherung eines öffentlichen Schlüssels und einer Identifikation für den öffentlichen Schlüssel, einem geschützten Speicherbereich zur Speicherung wenigstens eines Offsets und Verarbeitungsmitteln, die programmiert sind zur Entschlüsselung der verschlüsselten Datenblocks in einer derartigen empfangenen MPEG Tabelle durch den gespeicherten öffentlichen Schlüssel zu dem privaten Schlüssel, zum Nachschlagen des einen gespeicherten Offsets in einem geschützten Speicherbereich, zur Extrahierung der Signatur von dem entschlüsselten Datenblock durch den nachgeschlagenen Offset von dem Start des entschlüsselten Datenblocks, zur Erzeugung einer Signatur für die in der empfangenen MPEG Tabelle empfangenen Daten und zum Vergleich der aus dem entschlüsselten Datenblock extrahierten Signatur mit der beim Empfänger/Decoder für die empfangenen Daten erzeugten Signatur.

[0051] Der Speicher für die Speicherung des Offsets ist vorzugsweise durch einen neubeschreibbaren, nicht-flüchtigen Speicher gebildet.

[0052] In dem Empfänger/Decoder des neunten bis zwölften Aspekt der Erfindung können die Verarbeitungsmittel dafür programmiert sein, eine jeweilige Modulsignatur für jedes der Module in den empfangenen Modul-MPEG Tabellen und zum Vergleich jeder Modulsignatur in der empfangenen Verzeichnis-MPEG Tabelle mit der durch den Empfänger/Decoder erzeugten Modulsignatur.

[0053] Das letztgenannte Merkmal kann unabhängig von dem neunten bis zwölften Aspekt der Erfindung durchgeführt werden. Daher liefert ein dreizehnter Aspekt der Erfindung einen MPEG Empfänger/Decoder mit: Mitteln zum Empfang eines derartigen Verzeichnisses und Modul-MPEG Tabellen, Mitteln zur Speicherung eines öffentlichen Schlüssels und einer Identifikation für den öffentlichen Schlüssel und Verarbeitungsmitteln, die programmiert sind für die Entschlüsselung der verschlüsselten Verzeichnissignatur in der empfangenen Verzeichnis-MPEG Tabelle durch den öffentlichen Schlüssel entsprechend dem privaten Schlüssel, zur Erzeugung einer Verzeichnissignatur für das Verzeichnis in der empfangenen Verzeichnis-MPEG Tabelle, zum Vergleich der entschlüsselten Verzeichnissignatur und der durch den Empfänger/Decoder Verzeichnissignatur, zur Erzeugung einer jeweiligen Modulsignatur für jedes der Module in den empfangenen Modul-MPEG Tabellen und zum Vergleich jeder Modulsignatur in der empfangenen Verzeichnis-MPEG Tabelle mit der jeweiligen, durch den Empfänger/Decoder erzeugten Modulsignatur.

[0054] Die Verarbeitungsmittel sind vorzugsweise programmiert zur Verhinderung oder zum Abbruch des Herunterladens eines derartigen Datenmoduls, wenn die Modulsignatur in der empfangenen Verzeichnis-MPEG Tabelle und die jeweilige Modulsignatur, erzeugt bei dem Empfänger/Decoder für dieses Modul, nicht aneinander angepasst sind.

[0055] In dem Empfänger/Decoder eines neunten bis dreizehnten Aspekts der Erfindung werden die Verarbeitungsmittel vorzugsweise zur Verhinderung oder zum Abbruch des Herunterladens der Daten programmiert, wenn die oder jede entschlüsselte Signatur und die erzeugte Signatur nicht miteinander übereinstimmen.

[0056] Bevorzugte Merkmale der vorliegenden Erfindung werden nunmehr an einem Beispiel mit Bezug auf die beigefügte Zeichnung beschrieben. Darin:

[0057] Fig. 1 zeigt den Gesamtaufbau eines digitalen Fernsehsystems,

[0058] Fig. 2 zeigt den Aufbau eines interaktiven Systems des digitalen Fernsehsystems von Fig. 1,

[0059] Fig. 3 ist ein Schema von Schnittstellen eines den Teil des Systems von Fig. 1 und Fig. 2 bildenden Empfänger/Decoders,

[0060] Fig. 4 ist ein Schema einer in dem digitalen Fernsehsystem benutzten Fernbedienung,

[0061] Fig. 5 zeigt die Anordnung der Dateien in einem Modul, das in dem Speicher eines interaktiven Empfänger/Decoders heruntergeladen wird,

[0062] Fig. 6 zeigt einen Zusammenhang zwischen einer Zahl von Komponenten eines MPEG-Stroms,

[0063] Fig. 7 zeigt, wie eine Anwendung aus den Modulen/Tabellen aufgebaut werden kann, die daraufhin aus Abschnitten gebildet sein kann,

[0064] Fig. 8 zeigt den Inhalt eines Verzeichnismoduls,

[0065] Fig. 9 zeigt detaillierter einen Teil des Inhalts des Verzeichnismoduls, und

[0066] Fig. 10 zeigt verschiedene Bereiche eines Speichers in einem Empfänger/Decoder des Fernsehsystems.

[0067] Eine Übersicht eines digitalen Fernsehsystems **1000** gemäß der vorliegenden Erfindung ist in Fig. 1 dargestellt. Die Erfindung enthält ein weitestgehend bekanntes digitales Fernsehsystem **2000**, das das bekannte MPEG-2 Komprimiersystem benutzt, um komprimierte digitale Signale zu übertragen. Im Einzelnen empfängt der MPEG-2 Komprimierer **2002** in einem Sendezentrum einen digitalen Signalstrom (im Allgemeinen einen Strom von Videosignalen). Der Komprimierer **2002** ist über eine Strecke **2006** mit einem Multiplexer und Verwürfeler **2004** verbunden. Der Multiplexer **2004** empfängt mehrere weitere Eingangssignale, stellt einen oder mehrere Transportströme zusammen und überträgt komprimierte digitale Signale über eine Strecke **2010** zu einem Sender **2008** des Sendezentrums, die eine weite Vielfalt von Formen einschließlich Telekommunikationsstrecken annehmen kann. Der Sender **2008** überträgt elektromagnetische Signale über eine nach oben gerichtete Strecke **2012** zu einem Satellitentransponder **2014**, wo sie elektronisch verarbeitet und über eine fiktive, nach unten gerichtete Strecke **2016** zu einem erdgebundenen Empfänger **2018** übertragen werden, konventionell in der Form einer Schüssel, die der Endbenutzer besitzt oder gemietet hat. Die von dem Empfänger **2018** empfangenen Signale werden einem integrierten Empfänger/Decoder **2020** zugeführt, den der Endbenutzer besitzt oder gemietet hat und die mit dem Fernsehgerät **2022** des Endbenutzers verbunden ist. Der Empfänger/Decoder **2020** decodiert das komprimierte MPEG 2-Signal in ein Fernsehsignal für das Fernsehgerät **2022**.

[0068] Ein System **3000** für einen bedingten Zugriff ist mit dem Multiplexer **2004** und dem Empfänger/Decoder **2020** verbunden und liegt teilweise in dem Sendezentrum und teilweise in dem Decoder. Es ermöglicht dem Endbenutzer einen Zugriff zu digitalen Fernsehsendungen von einem oder mehreren Sendeanbietern. Eine Smart Card, die Nachrichten für kommerzielle Angebote entschlüsseln kann (das ist, eine oder mehrere durch die Sendeanbieter verkaufte Fernsehprogramme) kann in dem Empfänger/Decoder **2020** enthalten sein. Durch Anwendung des Decoders **2020** und der Smart Card kann der Endbenutzer kommerzielle Angebote in einem Abonnementbonus oder einem Gebührenfernsehmodus kaufen.

[0069] Ein interaktives System **4000**, das ebenfalls mit dem Multiplexer **2004** verbunden ist, und der Empfänger/Decoder **2020**, der wiederum teilweise in dem Sendezentrum und teilweise in dem Decoder liegt, befähigt den Endbenutzer, über einen Modem-Rückkanal **4002** mit verschiedenen Anwendungen zusammen zu arbeiten.

[0070] Fig. 2 zeigt den allgemeinen Aufbau des interaktiven Fernsehsystems **4000** des digitalen Fernsehsystems **1000** der vorliegenden Erfindung.

[0071] Zum Beispiel ermöglicht das interaktive System **4000** einem Endbenutzer, Programmpunkte von Bildschirmkatalogen, örtlichen Zeitschriften und Wetteraufstellungen nach Anforderung und über sein Fernsehgerät Spiele vorzunehmen.

[0072] Das interaktive System **4000** enthält in einer Übersicht vier Hauptelemente:

- Ein Berechtigungswerkzeug **4004** bei dem Sendezentrum oder woanders zur Ermöglichung eines Sendeanbieters, Anwendungen zu bilden, zu entwickeln, zu überprüfen und zu testen,
- einen Anwendungs- und Datenserver **4006** bei dem Sendezentrum, das mit dem Berechtigungswerkzeug **4004** verbunden ist, damit ein Sendeanbieter Anwendungen vorbereiten, berechtigen und formatieren kann, und Daten für die Lieferung zu dem Multiplexer und Verwürfeler **2004** zur Einfügung in den MPEG-2 Transportstrom (im Allgemeinen der private Bereich davon) zu dem End-

benutzer,

• eine virtuelle Maschine als Laufzeitmaschine (RTE) **4008**, die ein durchführbarer Code ist, der in dem Empfänger/Decoder **2020** angeordnet ist, den der Endbenutzer besitzt oder gemietet hat, damit ein Endbenutzer Anwendungen in den Arbeitsspeicher des Decoders **2020** für die Durchführung empfangen, berechtigen, dekomprimieren und laden kann. Das Gerät **4008** läuft außerdem selbstständig für Anwendungen mit einem allgemeinen Zweck. Das Gerät **4008** ist unabhängig von der Hardware und dem Betriebssystem, und

• einen Modem-Rückkanal **4002** zwischen dem Empfänger/Decoder **2020** und der Anwendung und dem Datenserver **4006**, damit Signale, die den Server **4006** informieren, Daten und Anwendungen in den MPEG-2 Transportstrom auf Anforderung zu dem Endbenutzer instruieren kann.

[0073] Das interaktive Fernsehsystem arbeitet mit "applications" (Anwendungen), die die Funktionen des Empfänger/Decoders und verschiedene darin enthaltenen Serviceleistungen steuert. Anwendungen sind in dem Gerät **4008** als "resource files" (Ressourcendateien) dargestellt. Ein "module" ist ein Satz von Ressourcendateien und Daten. Ein "Speichervolumen" des Empfänger/Decoders ist ein Speicherplatz für Module. Module können von dem MPEG-2 Transportstrom in den Empfänger/Decoder **2020** heruntergeladen werden.

[0074] Physische oder körperliche Schnittstellen des Empfänger/Decoders **2020** werden für das Herunterladen der Daten benutzt. In Fig. 3 enthält der Decoder **2020** zum Beispiel sechs Herunterladegeräte, einen MPEG Flusstuner **4028**, eine serielle Schnittstelle **4030**, eine parallele Schnittstelle **4032**, ein Modem **4034** und zwei Kartenleser **4036**.

[0075] Für die Zwecke dieser Beschreibung ist eine Anwendung ein Teil eines Computercodes für die Steuerung von High Level (hochwertigen) Funktionen von vorzugsweise dem Empfänger/Decoder **2020**. Wenn zum Beispiel der Endbenutzer den Fokus der Fernbedienung **2026** (wie detaillierter in Fig. 4 dargestellt ist) auf einen auf dem Schirm des Fernsehgeräts **2022** sichtbaren Gegenstand richtet und die Berechtigungstaste drückt, läuft die Instruktionsfolge mit dieser Taste ab.

[0076] Eine interaktive Anwendung schlägt Menüs vor und führt Befehle aus auf Anforderung durch den Endbenutzer und liefert Daten für den Zweck der Anwendung. Anwendungen können entweder residente Anwendungen sein, das heißt, gespeichert in dem ROM (oder FLASH oder einem anderen nicht-flüchtigen Speicher) des Empfänger/Decoders **2020** oder gesendet und in das RAM oder FLASH des Decoders **2020** heruntergeladen sein.

[0077] Beispiele von Anwendungen sind:

• Eine Auslöseanwendung. Der Empfänger/Decoder **2020** ist mit einer residenten Auslöseanwendung versehen, die eine adaptierbare Sammlung von Modulen sein kann (dieser Ausdruck wird später im Einzelnen definiert), der es dem Empfänger/Decoder **2020** ermöglicht, unverzüglich auf dem MPEG-2 Gebiet tätig zu sein. Die Anwendung liefert Kernmerkmale, die im Bedarfsfall durch den Senderanbieter modifiziert werden können. Sie bildet außerdem eine Schnittstelle zwischen residenten Anwendungen und heruntergeladenen Anwendungen.

• Eine Start-Anwendung. Die Start-Anwendung erlaubt es jeder Anwendung, entweder heruntergeladen oder resident auf dem Empfänger/Decoder **2020** abzulaufen. Diese Anwendung arbeitet als ein so genanntes Bootstrap, ausgeführt bei der Ankunft eines Service, um die Anwendung zu starten. Das Starten wird in ein RAM heruntergeladen und kann daher leicht aktualisiert werden. Es ist derart konfiguriert, dass die interaktiven Anwendungen, die auf jedem Kanal verfügbar sind, und entweder unmittelbar nach dem Herunterladen oder nach dem Vorladen gewählt werden können. Im Falle des Vorladens wird die Anwendung in den Speicher **2024** geladen und im Bedarfsfall durch das Starten aktiviert.

• Ein Programmführer. Der Programmführer ist eine interaktive Anwendung, die volle Informationen über das Programm bietet. Zum Beispiel kann er Informationen liefern über, zum Beispiel, die Fernsehprogramme einer Woche, die auf jedem Kanal eines digitalen Fernsehers geliefert werden. Durch Drücken einer Taste auf der Fernbedienung **2026** nimmt der Endbenutzer Zugriff zu einem zusätzlichen Schirm, überlagert auf dem in dem Schirm des Fernsehgeräts **2022** dargestellten Ereignis. Dieser zusätzliche Schirm ist ein Browser, der Informationen für das laufende und das nächste Ereignis jedes Kanals des digitalen Fernsehbouquets gibt. Durch Drücken einer anderen Taste auf der Fernbedienung **2026** nimmt der Endbenutzer Zugriff zu einer Anwendung, die eine Liste von Informationen über Ereignisse über eine Woche darstellt. Der Endbenutzer kann außerdem mit einfachen und ihm geläufigen Kriterien Ereignisse suchen und sortieren. Der Endbenutzer kann auch direkt zu einem gewählten Kanal Zugriff nehmen.

• Eine Gebührenfernsehanwendung. Die Gebührenfernsehanwendung ist ein interaktiver Service, der auf jedem PPV-Kanal des digitalen Fernsehbouquets in Verbindung mit dem System **300** für einen bedingten Zugriff verfügbar ist. Der Endbenutzer kann Zugriff nehmen zu der Anwendung durch einen Fernsehführer oder einen Kanal-Browser. Zusätzlich startet die Anwendung automatisch, sobald ein PPV-Ereignis auf dem PPV-Kanal ermittelt worden ist. Der Endbenutzer

ist dann in der Lage, das laufende Ereignis durch seine Tochter-Smart Card **3020** oder über den Kommunikationsserver **3022** zu kaufen (durch Anwendung eines Modem, eines Telefons und durch DTMF-Codes, MINITEL oder dergleichen). Die Anwendung kann entweder in dem ROM des Empfängers/Decoders **2020** resident sein oder kann in das RAM des Empfängers/Decoders **2020** heruntergeladen werden.

• Eine PC-Herunterlade-Anwendung. Auf Anforderung kann der Endbenutzer eine Computersoftware durch Anwendung der PC-Herunterlade-Anwendung herunterladen.

• Eine Zeitschrift Browser Anwendung. Die Zeitschrift Browser Anwendung enthält eine zyklische Videosendung von Bildern mit einer Endbenutzernavigation über so genannte "on-screen" Tasten.

• Eine Quiz- oder Frageanwendung. Die Quizanwendung ist vorzugsweise synchronisiert mit einem gesendeten Quizprogramm. Als ein Beispiel werden mehrere Wahlfragen auf dem Schirm des Fernsehgeräts **2022** wiedergegeben, und der Endbenutzer kann durch Anwendung der Fernbedienung **2026** eine Antwort wählen. Die Quizanwendung kann den Benutzer informieren, ob die Antwort richtig ist oder nicht, und kann die Punkte des Benutzers halten.

• Eine "Teleshopping" (Ferneinkauf)-Anwendung. In einem Beispiel der Teleshopping-Anwendung werden Angebote der Verkaufsgüter zu dem Empfänger/Decoder **2020** übertragen und auf dem Fernsehgerät **2022** wiedergegeben. Durch Anwendung der Fernbedienung kann der Benutzer einen bestimmten Gegenstand für das Kaufen wählen. Die Order für den Gegenstand wird über den Modem-Rückkanal **4002** zu der Anwendung und dem Datenserver **4006** oder zu einem getrennten Verkaufssystem übertragen, dessen Telefonnummer zu dem Empfänger/Decoder heruntergeladen wurde, möglicherweise mit einer Order, eine Kreditkarte zu belasten, die in einen der Kartenleser **4036** des Empfänger/Decoders **2020** eingeschoben wurde.

• Eine Telebanking-Anwendung. In einem Beispiel der Telebanking-Anwendung gibt der Benutzer eine Bankkarte in einen der Kartenleser **4036** des Empfänger/Decoders **2020** ein. Der Empfänger/Decoder **2020** ruft die Bank des Benutzers an, durch Anwendung einer Telefonnummer, die in der Bankkarte oder in dem Empfänger/Decoder gespeichert ist, und dann liefert die Anwendung eine Zahl von Möglichkeiten, die mit der Fernbedienung **2026** gewählt werden können, zum Beispiel zum Herunterladen über die Telefonleitung eines Bankauszugs, die Kontostände zwischen Konten überträgt, ein Scheckbuch anfordert, usw..

• Eine Internet-Browser-Anwendung. In einem Beispiel der Internet-Browser-Anwendung werden Anweisungen von dem Benutzer, wie eine Anforderung zur Betrachtung einer Webseite mit einer bestimmten URL, durch Anwendung der Fernbedienung **2026** eingegeben, und diese werden durch den Modem-Rückkanal **4002** zu der Anwendung und dem Datenserver **4006** übertragen. Die richtige Webseite wird dann durch das Sendezentrum in die Übertragungen durch den Empfänger/Decoder **2020** über die nach oben gerichtete Strecke **2021** aufgenommen, den Transponder **2014** und die nach unten gerichtete Strecke **2016** empfangen und auf dem Fernsehgerät **2022** wiedergegeben.

[0078] Anwendungen werden in Speicherplätzen in dem Empfänger/Decoder **2020** gespeichert und als Ressourcendateien dargestellt. Die Ressourcendateien enthalten graphische oder Objekt-Beschreibungs-Dateien, variable Block-Einheits-Dateien, Instruktionsfolge-Dateien, Anwendungsdateien und Datendateien.

[0079] Die Beschreibungseinheit für die graphischen Objekte beschreiben die Schirme, die Mensch/Maschine-Schnittstelle der Anwendung. Die variablen Bock-Einheits-Dateien beschreiben die durch die Anwendung verarbeiteten Datenstrukturen. Die Anweisungs-Folgedateien beschreiben die Verarbeitungsvorgänge der Anwendungen. Die Anwendungsdateien bilden die Eingangspunkte für die Anwendungen.

[0080] Die auf diese Weise gebildeten Anwendungen können Datendateien benutzen, wie Icon-Bibliotheksdateien, Bilddateien, Charakterdateien, Farbtabellendateien und ASCII-Text-Dateien. Eine interaktive Anwendung kann außerdem Online-Daten durch Bildung von Eingängen oder Ausgängen enthalten.

[0081] Das Gerät **4008** lädt in seinem Speicher diejenigen Ressourcendateien, die es zu einer bestimmten Zeit benötigt. Diese Ressourcendateien werden aus den graphischen Objekt-Beschreibungs-Dateieinheiten, Instruktions-Folge-Dateien und Anwendungsdateien gelesen. Variable Block-Einheits-Dateien werden in dem Speicher gespeichert, der auf einen Anruf zu einem Vorgang zum Laden der Module folgt und bleiben dort verriegelt, bis ein spezifischer Anruf für einen Vorgang für das Entladen oder Löschen von Modulen erfolgt.

[0082] In Fig. 3 sind ein Modul **4010**, wie ein Teleshopping-Modul, ein Satz Ressourcendateien und Daten dargestellt, die Folgendes enthalten:
eine einzelne Anwendungsdatei **4012**,
eine unbestimmte Zahl von Beschreibungs-Dateieinheiten für graphische Objekte **4014**,
eine unbestimmte Zahl von variablen Block-Einheits-Dateien **4016**,
eine unbestimmte Zahl von Instruktions-Folge-Dateien **4018** und
wo angebracht, Datendateien **4020**, wie Icon-Biblio-

theks-Dateien, Bilddateien, Charakter-Dateien, Tabellen-Dateien und ASCII-Text-Dateien.

[0083] In Fig. 5 ist ein Modul **4010**, wie ein Teleshopping-Modul, ein Satz von Ressourcendateien und Daten, die Folgendes enthalten:
eine einzige Anwendungsdatei **4012**,
eine unbestimmte Zahl von Beschreibungs-Dateieinheiten für graphische Objekte **4014**,
eine unbestimmte Zahl von Einheitsdateien **4016** für variable Blöcke,
eine unbestimmte Zahl von Instruktions-Folge-Dateien **4018** und
wo angebracht, Datendateien **4020**, wie die Icon-Bibliotheks-Dateien, Bilddateien, Eigenschafts-Dateien, Farbtabellen-Dateien und ASCII-Text-Dateien.

[0084] Das Konzept der Module **4010** zusammen mit dem Konzept des Herunterladens von kleinen Codeteilen ermöglicht die einfache Weiterentwicklung von Anwendungen. Sie können in den Festspeicher FLASH des Decoders heruntergeladen werden als residente Software oder gesendet werden, um nur dann in das RAM des Decoders **2020** heruntergeladen werden, wenn sie durch den Endbenutzer benötigt wird.

[0085] Zum Herunterladen eines Moduls **4010** von einem Trägersignal wird zunächst ein auf dem Trägersignal zugängliches Verzeichnis heruntergeladen. Dieses Verzeichnis enthält einfach die Namen der Module **4010**, die von dem Trägersignal heruntergeladen werden können. Wenn dieses Verzeichnis heruntergeladen worden ist, ist es für die Anwendung möglich, eines oder mehrere Module **4010** herunterzuladen. In dem Fall eines MPEG-Flusses wird das Verzeichnis in eine einzige MPEG Tabelle überführt. Außerdem wird ein Modul **4010** in eine einzige MPEG Tabelle überführt. In dem Fall von zu dem MPEG Tuner **4028** übertragenen Modulen wird das lange MPEG-2 Format benutzt mit einem langen Header und einem CRC-Code. Das ist auch der Fall bei den fünf anderen Schnittstellen (serielle Schnittstelle **4030**, parallele Schnittstelle **4032**, Modem **4034** und zwei Kartenleser **4036**), ausgenommen, wenn das "kurze" MPEG-2 Format mit einem kürzeren Header und keine CRC benutzt wird.

[0086] Insbesondere enthält in Fig. 6, wie es bekannt ist, der MPEG-2-Bitstrom eine Programmzugriffstabelle ("PAT") **10** mit einer Paketidentifikation ("PID") von 0. Die PAT enthält Hinweise auf die PIDs der Programm-Map-Tabellen ("PMTs") **12** einer Zahl von Programmen. Jede PMT enthält einen Hinweis auf die PIDs der Ströme der Audio-MPEG Tabellen **14** und Video-MPEG Tabellen **16** für dieses Programm. Ein Paket mit einem PID von null, das ist die Programmzugriffstabelle **10**, bildet den Eingangspunkt für alle MPEG-Zugriffe.

[0087] Zum Herunterladen von Anwendungen und Daten für diese werden zwei neue Stromtypen definiert, und die relevante PMT enthält ebenfalls Hinweise auf die PIDs der Ströme von Anwendungs-MPEG-Tabellen **18** (oder Abschnitte von diesen) und Daten-MPEG Tabellen **20** (oder Abschnitte von diesen).

[0088] In Fig. 7 ist, um eine Anwendung **22** herunterzuladen, die Anwendung in Module **24** aufgeteilt und jedes durch eine MPEG-Tabelle gebildet, von denen einige durch einen einzelnen Abschnitt **18** und von denen andere durch mehrere Abschnitte **18** gebildet werden. Ein üblicher Abschnitt **18** enthält einen Header **26**, der eine Ein-Bit-Tabellenidentifikation ("TID) **28**, die Abschnittsnummer **30** dieses Abschnitts in der Tabelle, die Gesamtzahl **32** der Abschnitte in dieser Tabelle und eine Zwei-Byte-TID-Erweiterung **34** enthält. Jeder Abschnitt enthält außerdem den Datenteil **36** und eine CRC **38**. Für eine bestimmte Modul/Tabelle **24** haben alle die Abschnitte **18** bildenden Datentabelle **24** dieselbe TID **28** und dieselbe TID-Erweiterung **34**. Für eine bestimmte Anwendung **22** haben alle diese Anwendung **22** bildenden Tabellen **24** dieselbe TID **28**, jedoch unterschiedliche jeweilige TID-Erweiterungen.

[0089] Für jede Anwendung **22** gibt es eine einzige derartige MPEG Tabelle **24**, die als Verzeichnis dient, und die detaillierter in Fig. 8 dargestellt ist. Die Verzeichnistabelle **40** enthält einen Header **26**, einen Verzeichnisteil **42**, eine Schlüssel-Identifikation **44**, eine verschlüsselte Signatur **46** und eine CRC **38**. Aus dem Obigen ergibt sich, dass die Verzeichnistabelle **40** in ihrem Header **26** dieselbe TID **28** wie die anderen, die Anwendung bildenden Module/Tabellen **24** hat. Jedoch hat die Verzeichnistabelle eine vorbestimmte TID-Erweiterung **34** von null, und alle anderen Module **24** haben Nicht-Null-TID-Erweiterungen. Der Header enthält außerdem eine Versionsnummer **48** für die Verzeichnistabelle **40**. Der Verzeichnisteil **42** enthält für jede der die Anwendung bildenden anderen Modul/Tabellen **24** den Namen **50** des Moduls, die TID-Erweiterung **34** für jedes Modul und eine Signatur **52** dieses Moduls. Der Verzeichnisteil **42** kann auch für jede der anderen Modul/Tabellen **24** die Länge dieses Moduls und die Versionsnummer des Moduls enthalten.

[0090] Wieder zu Fig. 6: Im Betrieb werden die PAT **10**, PMTs **12** und die Anwendungs- und Datenstromkomponenten **18**, **20** zyklisch übertragen und nötigenfalls aktualisiert. Jede Anwendung, die übertragen wird, hat eine jeweilige vorbestimmte TID **28**. Zum Herunterladen einer Anwendung wird die MPEG-Tabelle mit der richtigen TID und einer TID-Erweiterung von null zu dem Empfänger/Decoder **2020** heruntergeladen. Das ist daher die Verzeichnistabelle **40** für die benötigte Anwendung. Die Daten in dem Verzeichnis werden dann durch den Empfänger/De-

coder **2020** verarbeitet, um die TID-Erweiterungen **34** der Modultabellen zu ermitteln, die die geforderte Anwendung bilden, und dann kann jede benötigte Modultabelle mit derselben TID wie die Verzeichnistabelle und eine von dem Verzeichnis gelieferte TID-Erweiterung heruntergeladen werden.

[0091] Der Empfänger/Decoder **2020** dient zur Prüfung der Verzeichnistabelle für jede Aktualisierung derselben. Das kann erfolgen durch Herunterladen der Verzeichnistabelle, wieder periodisch, zum Beispiel alle 30 Sekunden oder alle fünf Minuten und Vergleich der Versionsnummer der frisch herunter geladenen Verzeichnistabelle mit der Versionsnummer der vorher herunter geladenen Verzeichnistabelle. Wenn die frisch heruntergeladene Versionsnummer später ist, dann werden die Module für die vorherige Verzeichnistabelle oder andere derartige Module, für die spätere Versionsnummer bestehen, ausgehängt (unmounted), und die späteren Module werden heruntergeladen und eingehängt. In einer alternativen Anordnung wird der ankommende Bitstrom durch eine Maske gefiltert, entsprechend der TID, TID-Erweiterung und Versionsnummer mit für die TID der Anwendung gesetzten Werte, und eine TID-Erweiterung von null und einer Versionsnummer eins größer als die Versionsnummer des herunter geladenen Verzeichnisses. Daher kann ein Inkrement der Versionsnummer detektiert werden, und nach der Detektion wird das Verzeichnis heruntergeladen und die Anwendung aktualisiert, oben beschrieben. Eine weitere Beschreibung einer derartigen Filterung ist enthalten in der prioritätsgleichen Anmeldung (Bezugszeichen beim Anwalt Nr. PDC/ASB/19716). Wenn eine Anwendung terminiert werden soll, wird ein leeres Verzeichnis mit der nächsten Versionsnummer übertragen, jedoch ohne jedes der in dem Verzeichnis aufgelisteten Module. In der Antwort auf den Empfang eines derartigen leeren Verzeichnisses wird der Empfänger/Decoder **2020** zum Aushängen (unmount) der Anwendung programmiert.

[0092] Die Anwendung der Signaturen und die Verschlüsselung für die Anwendungstabellen werden nunmehr im Detail beschrieben.

[0093] Wie oben beschrieben, enthält die Eingabe für jedes Modul in die Verzeichnistabelle **40** die Modulsignatur. Die Modulsignatur wird erzeugt durch Anwendung einer bekannten MD5-Signatur Erzeugungsvorgang auf den Daten in der jeweiligen Modultabelle.

[0094] Außerdem enthält die Verzeichnistabelle **40** eine verschlüsselte Signatur **46**, die in der Weise erzeugt wird, die nunmehr anhand der Fig. 9 beschrieben wurde. Es wird ein Block **54** von 64 Byte von Daten erzeugt. Das erste Byte **56** ist null. Die nächsten drei Byte **58** können Dummy- oder willkürliche Daten enthalten. Die nächsten acht Byte **60** bilden eine Anwendungs-Validations-Bitmap, die im Folgenden näher beschrieben wird. Die letzten vier Byte **62** werden reserviert. Die übrigen 32 Byte enthalten eine 16 Byte-Signatur **64**, die bei einem Offset zwischen 0 und 31 Byte beginnt, nach dem ersten Byte, das auf die Anwendungs-Validations-Bitmap **60** folgt. Die Dummy-Daten werden zwischen der Anwendungs-Validations-Bitmap **60** und der Signatur **64** und/oder zwischen der Signatur **64** und dem reservierten Byte **62** eingefügt. Die Signatur **64** wird erzeugt durch den bekannten MD5 Signaturerzeugungsvorgang auf den Verzeichniseingängen **42** in der Verzeichnistabelle **40**. Der Block **54** wird dann durch Anwendung eines bekannten Verschlüsselungsvorgangs und eines bestimmten privaten Schlüssels zur Erzeugung der verschlüsselten Signatur und der Anwendungs-Validations-Bitmap **46** verschlüsselt. Dieser Datenblock ist in der Verzeichnistabelle **40** enthalten, und eine 1-Byte-Identifikation des privaten Schlüssels, der benutzt wurde zur Verschlüsselung des Blocks, ist als die Schlüsselidentifikation **44** in der Verzeichnistabelle **40** enthalten.

[0095] Zur Zusammenfassung der Erzeugung einer Anwendung und ihrer Übertragung werden die folgenden Schritte durchgeführt:

- Erzeugung der Anwendung als mehrere Module,
- Notierung der vorbestimmten TID **28** für die Anwendung,
- Zuordnung von Namen und Nicht-Null TID-Erweiterungen **34** für die Module,
- Formatierung jedes Moduls als eine MPEG Tabelle **24** oder Abschnitte **18** von einer MPEG Tabelle,
- Erzeugung des Verzeichnisses **42**,
- Erzeugung einer MD5 Signatur **64** für das Verzeichnis,
- Wahl eines Anwendungs-Validations-Bit **60**,
- Wahl eines Offset,
- Erzeugung des Blocks **54**,
- Verschlüsselung des Blocks **54** durch eine Verschlüsselung mit einem gewählten privaten Schlüssel,
- Erzeugung der Verzeichnis-MPEG Tabelle **40** mit der zugeordneten TID **28**, einer TID-Erweiterung von null, das Verzeichnis **42**, eine Identifikation **44** des privaten Schlüssels und der verschlüsselten Signatur **46**,
- Übertragung der Verzeichnistabelle **40** und der Modultabellen **24** oder Abschnitte **18**.

[0096] Der Betrieb des Empfänger/Decoders **2020** in der Behandlung mit Signaturen und der Entschlüsselung während des Herunterladens einer Anwendung werden nunmehr beschrieben. In Fig. 10 enthält der Enmpfänger/Decoder **2020** ein EEPROM **68**, ROM **70** und RAM **72**. Das EEPROM **68** enthält einen geschützten Bereich **74**, der durch die virtuelle Maschine benutzt wird und wo nur die virtuelle Maschine (und nicht eine normale Anwendung) schrei-

ben kann. Der geschützte Abschnitt **74** enthält eine Schlüssel-Validations-Bitmap **76** von 16 oder 256 Bit, eine Anwendungs-Validations-Bitmap **78** mit 64 Bit und eine Offset-Bitmap **80** von 32 Bit. Das ROM **70** enthält in einer Ausführungsform sechzehn öffentliche Schlüssel **82**. In diesem Fall wird eine 16 Bit Schlüssel-Validations-Bitmap angewendet und in einer anderen Ausführung **256** öffentliche Schlüssel, in welchem Fall eine 256 Bit-Schlüssel-Validations-Bitmap angewendet wird. Die öffentlichen Schlüssel werden durch ihre physische oder räumliche Lage in dem RAM **70** identifiziert, oder sie können alternativ in einer Lookup-Tabelle enthalten sein, wobei eine besondere Schlüsselidentifikation den entsprechenden öffentlichen Schlüssel enthalten kann. Das RAM **72** kann dazu dienen, einen vorübergehenden Schlüssel **84** zu speichern.

[0097] Wie oben erwähnt, wird, wenn eine Anwendung heruntergeladen werden soll, zunächst die Verzeichnistabelle mit der vorbestimmten TID für diese Anwendung und eine TID-Erweiterung von null heruntergeladen. Die Schlüsselidentifikation **44** wird dann aus der Verzeichnistabelle extrahiert, und es erfolgt eine Prüfung des Schlüssel-Validations-Bitmaps **76** in dem geschützten Speicher **74**, das das der extrahierten Schlüsselidentifikation **44** entsprechende Bit gesetzt ist. Wenn dies nicht der Fall ist, dann wird das weitere Herunterladen der Anwendung abgebrochen. Wenn jedoch der richtige Schlüssel gesetzt ist, dann wird ein öffentlicher Schlüssel **82** aus dem ROM **70** gewählt, entsprechend der Identifikation **44** für den extrahierten Schlüssel. Der gewählte öffentliche Schlüssel und ein bekannter Entschlüsselungsvorgang dienen dann zur Entschlüsselung des verschlüsselten Blocks **46** in der Verzeichnistabelle **40** zur Erzeugung eines Blocks **54**. Die Anwendungs-Validations-Bitmap **60** wird dann aus dem entschlüsselten Block **54** extrahiert und AND-verknüpft mit der in den gespeicherten Anwendungs-Validations-Bitmap **78**. Wenn das Ergebnis des AND-Vorgangs null ist, wird ein weiteres Herunterladen der Anwendung abgebrochen. Wenn jedoch das Ergebnis des AND-Vorgangs nicht null ist, dann wird der Offset, der in der Offset-Bitmap **38** in dem geschützten Speicher **54** nachgeschlagen, und wenn mehr als ein Offset-Bit gesetzt ist, dann wird jedes Offset-Bit danach nachgeschlagen, und sechzehn Byte von Daten werden aus dem entschlüsselten Block **54** extrahiert, beginnend mit dem Nachschlage-Offset von dem ersten Byte nach der Anwendugns-Validations-Bitmap **60**. Für jeden Nachschlage-Offset werden die 16 Byte als Signatur in die Verzeichnistabelle **40** übertragen. Die Signatur der Eingänge in das Verzeichnis **42** der Verzeichnistabelle **40** wird durch den bekannten MD5 Vorgang berechnet, und diese berechnete Signatur wird verglichen mit der im Block **54** extrahierten Signatur. Wenn die beiden Signaturen für den oder jeden Nachschlage-Offset nicht übereinstimmen, dann wird das weitere Herunterladen der Anwendung abgebrochen. Wenn jedoch eine der Signaturen übereinstimmt, dann kann das Herunterladen der Module in das Verzeichnis **42** weiterlaufen. Wie oben erwähnt, wird, um ein bestimmtes Modul herunterzuladen, die TID-Erweiterung für dieses Modul von dem Verzeichnis **42** geliefert, und die MPEG Tabelle **24** oder Abschnitte **18** mit derselben TID wie die Verzeichnistabelle mit der aus dem TID gewonnenen Erweiterung wird heruntergeladen. Sobald die Modul-MPEG-Tabelle heruntergeladen worden ist, berechnet der Empfänger/Decoder **2020** die Signatur der heruntergeladenen Tabelle durch Anwendung des bekannten MD5 Vorgangs und vergleicht dann die berechnete Signatur mit der in der Verzeichniseingabe enthaltenen Signatur. Wenn die Signaturen übereinstimmen, dann wird das Modul akzeptiert, wenn sie jedoch nicht übereinstimmen, dann wird das Modul zurückgewiesen oder unterdrückt.

[0098] Alle Module der Anwendung können so in derselben spezifischen Weise wie oben heruntergeladen werden, und die Anwendung kann durch den Empfänger/Decoder **2020** laufen.

[0099] Mit den beschriebenen Merkmalen in dem Herunterladevorgang, die normalerweise benutzt werden, folgt nunmehr eine Beschreibung einiger Merkmale, die in dem letzten des Empfängers/Decoders **2020** und Änderung seiner Einstellungen benutzt werden.

[0100] Der Empfänger/Decoder **2020** ist so programmiert, dass der geschützte Speicherbereich **74** geändert werden kann, jedoch nur durch eine Anwendung, die durch eine bestimmte von Schlüsselidentifikationen heruntergeladen worden ist, zum Beispiel der Schlüssel **15** und mit einem bestimmten Offset, zum Beispiel ein Offset von null Byte, von dem ersten Byte nach der Anwendungs-Validations-Bitmap **60**. Der geschützte Speicher **74** kann geändert werden müssen, zum Beispiel, wenn zwei Operatoren, die denselben öffentlichen Schlüssel benutzt haben, sich entscheiden, dass sie verschiedene öffentliche Schlüssel benutzen möchten, oder wenn der Inhalt eines privaten Schlüssels entdeckt wurde, in welchem Fall der entsprechende öffentliche Schlüssel in der Schlüssel-Validations-Bitmap **76** als ungültig markiert wurde.

[0101] Der Empfänger/Decoder **2020** kann derart ausgebildet sein, dass einer der Schlüssel, zum Beispiel der Schlüssel **15**, immer verfügbar ist, in welchem Fall dieser Schlüssel kein Bit in der Schlüssel-Validations-Bitmap **76** benötigt. Daher kann dieses Bit für einen anderen Zweck benutzt werden. Insbesondere kann eine Anwendung, die durch den Schlüssel **15** authentifiziert worden ist, dafür vorgesehen sein, dieses bestimmte Bit auf 1 zu setzen, in welchem Fall der Empfänger/Decoder **2020** dafür programmiert ist, dass ein vorübergehender Schlüs-

sel **84** in das RAM **72** geladen werden kann, jedoch nur über die serielle Schnittstelle **4030**, die parallele Schnittstelle **4032** oder einen der beiden Kartenleser **4036**. Diese Möglichkeit kann zum Beispiel durch einen Hersteller des Empfänger/Decoders **2020** benutzt werden, der eine Anwendung empfangen kann dafür, dass ein vorübergehender Schlüssel in dem Empfänger/Decoder **2020** geladen werden kann, so dass er geprüft werden kann.

[0102] Die oben beschriebene Verschlüsselung und Signieranordnung bildet eine Anzahl von wichtigen Merkmalen. Insbesondere:

- Eine Anwendung kann nur heruntergeladen werden, wenn der Empfänger/Decoder **2020** den in seinem Speicher gespeicherten richtigen Schlüssel hat, entsprechend der Schlüsselidentifikation **44** in der heruntergeladenen Verzeichnistabelle,
- Für alle außer einem der Schlüssel kann eine Anwendung nur durch einen bestimmten Schlüssel heruntergeladen werden, wenn die Schlüssel-Validations-Bitmap **76** in dem Speicher des Empfängers/Decoders **2020** so gesetzt ist, dass dieser Schlüssel benutzt werden kann,
- eine Anwendung kann nur dann heruntergeladen werden, wenn ein gesetztes Bit in der Offset-Bitmap **28**, das in dem Speicher des Empfängers/Decoders **2020** gespeichert wird, dem Offset entspricht, der bei der Erzeugung der Verzeichnistabelle benutzt wird,
- eine Anwendung kann nur dann heruntergeladen werden, wenn die Anwendungs-Validations-Bitmap **78** in dem Speicher des Empfänger/Decoders **2020** derart richtig gesetzt ist, dass die Anwendung heruntergeladen werden kann,
- eine Anwendung kann nur dann heruntergeladen werden, wenn die Verzeichnistabelle nicht zerstört worden ist, nachdem ihre Signatur ursprünglich erzeugt wurde,
- jedes Modul einer Anwendung kann nur dann heruntergeladen werden, wenn die jeweilige Modultabelle nicht beschädigt worden ist, nachdem ihre Signatur ursprünglich erzeugt wurde,
- es wird nur ein Verschlüsselungsvorgang benötigt in der Vorbereitung einer Anwendung für das Herunterladen, obwohl die Anwendung aus mehreren MPEG Tabellen besteht, und nur ein Entschlüsselungsvorgang wird beim Empfänger/Decoder **2020** benötigt, um die vollständige Anwendung herunterzuladen,
- es können mehrere Schlüssel angewendet werden, so dass verschiedene Serviceanbieter verschiedene private Schlüssel haben können,
- ein vorübergehender Schlüssel kann zum Beispiel durch einen Hersteller für Prüfzwecke benutzt werden.

[0103] Natürlich wurde die vorliegende Erfindung oben nur an einem Beispiel beschrieben, und Änderungen von Details können innerhalb des Schutzumfangs der Erfindung vorgenommen werden.

[0104] In den obengenannten, bevorzugten Ausführungsformen wurden bestimmte Merkmale der vorliegenden Erfindung durch eine Computersoftware ausgeführt. Jedoch ist es natürlich für den Fachmann klar, dass jedes dieser Merkmale durch Hardware ausgeführt werden kann. Außerdem ist leicht verständlich, dass die durch die Hardware durchgeführten Funktionen, die Computersoftware und dergleichen durch Anwendung von elektrischen und ähnlichen Signalen durchgeführt werden können.

[0105] Es erfolgt ein Hinweis auf unsere anhängigen Anmeldungen, alle mit demselben Anmeldetag und bezeichnet mit Signal Generation and Broadcasting (Bezugszeichen beim Anwalt Nr. PC/ASB/19707), Smart Card für die Anwendung mit einem Empfänger von Encrypted Broadcast Signals und Empfänger (Aktenzeichen beim Anwalt Nr. PC/ASB/19708), Broadcast and Reception System and Conditional Access System dafür (Aktenzeichen beim Anwalt Nr. PC/ASB/19710), Downloading a Computer File from a Transmitter über einen Empfänger/Decoder zu einem Computer (Aktenzeichen beim Anwalt Nr. PC/ASB/19711), Transmission and Reception of Television Programmes and Other Data (Aktenzeichen beim Anwalt Nr. PC/ASB/19712), Downloading Data (Aktenzeichen beim Anwalt Nr. PC/ASB/19713), Computer Memory Organization (Aktenzeichen beim Anwalt Nr. PC/ASB/19714), Television or Radio Control System Development (Aktenzeichen beim Anwalt Nr. PC/ASB/19715), Extracting Data Sections from a Transmitted Data Stream (Aktenzeichen beim Anwalt Nr. PC/ASB/19716), Access Control System (Aktenzeichen beim Anwalt PC/ASB/19717), Data Processing System (Aktenzeichen beim Anwalt Nr. PC/ASB/19718) und Broadcast and Reception System und Receiver/Decoder and Remote Controller therefore (Aktenzeichen beim Anwalt Nr. PC/ASB/19720). Die Liste der Anmeldungen enthält die vorliegende Anmeldung.

Patentansprüche

1. Verfahren zum Herunterladen von mehreren Datenmodulen zu einem MPEG-Empfänger/Decoder (**2020**) mit folgenden Schritten:
Erzeugen einer Modulsignatur (**52**) für jedes herunterzuladende Datenmodul (**24**),
Formatierung der Datenmodule als jeweilige Modul-MPEG-Tabellen (**24**),
Erzeugen eines Verzeichnisses (**40**) mit einer Identifikation jeder Modul-MPEG-Tabelle (**24**) und der jeweiligen Signatur,
Erzeugen einer Verzeichnissignatur (**46**) für das Verzeichnis,
Verschlüsselung der Verzeichnissignatur in einem Datenblock (**66**) mit anderen Daten mit einem gewählten Offset zwischen dem Start des Datenblocks

und dem Start der Signatur durch einen privaten Schlüssel,
Formatierung des Verzeichnisses und der verschlüsselten Verzeichnissignatur als eine Verzeichnis-MPEG-Tabelle,
Übertragung des Verzeichnisses und der Modul-MPEG-Tabellen und
beim Empfänger/Decoder:
Empfang des Verzeichnisses und der Modul-MPEG-Tabellen,
Entschlüsselung des verschlüsselten Datenblocks mit der Verzeichnissignatur in der empfangenen Verzeichnis-MPEG-Tabelle durch einen dem privaten Schlüssel entsprechenden öffentlichen Schlüssel (**82**),
Nachschlagen wenigstens eines gespeicherten Offsets in einem geschützten Bereich des Speichers (**68**) des Empfänger/Decoders,
Extrahieren der Verzeichnissignatur von dem entschlüsselten Datenblock durch den nachgeschlagenen Offset von dem Start des entschlüsselten Datenblocks,
Erzeugen einer Verzeichnissignatur für das Verzeichnis in der empfangenen Verzeichnis-MPEG-Tabelle,
Vergleich der extrahierten Verzeichnissignatur mit der bei dem Empfänger/Decoder erzeugten Verzeichnissignatur,
Erzeugen einer jeweiligen Modulsignatur für jedes der Module in den empfangenen Modul-MPEG-Tabellen und
Vergleich jeder Modulsignatur in der empfangenen Verzeichnis-MPEG-Tabelle mit der jeweiligen, durch den Empfänger/Decoder erzeugten Modulsignatur.

2. Verfahren nach Anspruch 1, wobei der geschützte Speicherbereich wenigstens zwei derartige gespeicherte Offsets aufweist, wenn in dem Vergleichsschritt die extrahierte Signatur und die erzeugte Signatur nicht übereinstimmen, ferner mit den Schritten der Wiederholung des Nachschlagens, Extrahierung und Vergleich der Schritte durch einen anderen der gespeicherten Offsets.

3. Verfahren nach Anspruch 1 oder 2, wobei wenigstens einige der anderen Daten in dem Datenblock Dummy- oder willkürliche Daten sind.

4. Verfahren nach Anspruch 1, ferner mit dem Schritt der Verhinderung oder Abbrechung der Herunterladung eines derartigen Datenmoduls, wenn in dem Vergleichschritt für die Modulsignatur die Modulsignatur in der empfangenen Verzeichnis-MPEG-Tabelle und die bei dem Empfänger/Decoder für dieses Modul erzeugte Modulsignatur für dieses Modul nicht übereinstimmen.

5. Verfahren nach einem der Ansprüche 1 bis 4 mit dem Schritt der Verhinderung oder des Abbrechens der Herunterladung, wenn in dem Vergleichsschritt die entschlüsselte Verzeichnissignatur und die erzeugte Verzeichnissignatur nicht übereinstimmen.

6. MPEG Empfänger/Decoder (**2020**) zum Herunterladen von mehreren Datenmodulen (**24**) mit:
Mitteln zum Empfang des Verzeichnisses und der Modul-MPEG-Tabellen,
Mitteln zum Speichern eines öffentlichen Schlüssels (**82**) und einer Identifikation für den öffentlichen Schlüssel, Verarbeitungsmitteln für
Entschlüsselung eines Datenblocks mit einer Verzeichnissignatur (**46**), die mit einem privaten Schlüssel in der empfangenen Verzeichnis-MPEG-Tabelle verschlüsselt ist, durch den dem privaten Schlüssel entsprechenden öffentlichen Schlüssel,
Extrahieren der Signatur (**64**) von dem entschlüsselten Datenblock,
Erzeugen einer Verzeichnissignatur für das Verzeichnis in der empfangenen Verzeichnis-MPEG-Tabelle,
Vergleich der entschlüsselten Verzeichnissignatur und der bei dem Empfänger/Decoder erzeugten Verzeichnissignatur,
Erzeugen einer jeweiligen Modulsignatur für jedes der Module in den empfangenen Modul-MPEG-Tabellen und
Vergleich jeder Modulsignatur in der empfangenen Verzeichnis-MPEG-Tabelle mit der bei dem Empfänger/Decoder erzeugten jeweiligen Modulsignatur (**52**),
gekennzeichnet durch einen geschützten Speicherbereich zum Speichern wenigstens eines Offset (**80**) und dadurch, dass die Verarbeitungsmittel außerdem Mittel zum Nachschlagen wenigstens eines in dem geschützten Speicherbereich gespeicherten Offsets enthalten und durch Anwendung des einen nachgeschlagenen Offsets aus dem Start des entschlüsselten Datenblocks, um so die Signatur aus dem entschlüsselten Datenblock zu extrahieren.

7. Empfänger/Decoder nach Anspruch 6, wobei die Verarbeitungsmittel dafür programmiert sind, ein derartiges Datenmodul zu verhindern oder abzubrechen, wenn die Modulsignatur in der empfangenen Verzeichnis-MPEG-Tabelle und die bei dem Empfänger/Decoder für dieses Modul erzeugte jeweilige Modulsignatur nicht übereinstimmen.

8. Empfänger/Decoder nach Anspruch 6 oder 7, wobei die Verarbeitungsmittel dafür programmiert sind, die Daten zu sperren oder abzubrechen, wenn die Verzeichnissignatur und die erzeugte Verzeichnissignatur nicht miteinander übereinstimmen.

Es folgen 9 Blatt Zeichnungen

Anhängende Zeichnungen

Fig.1.

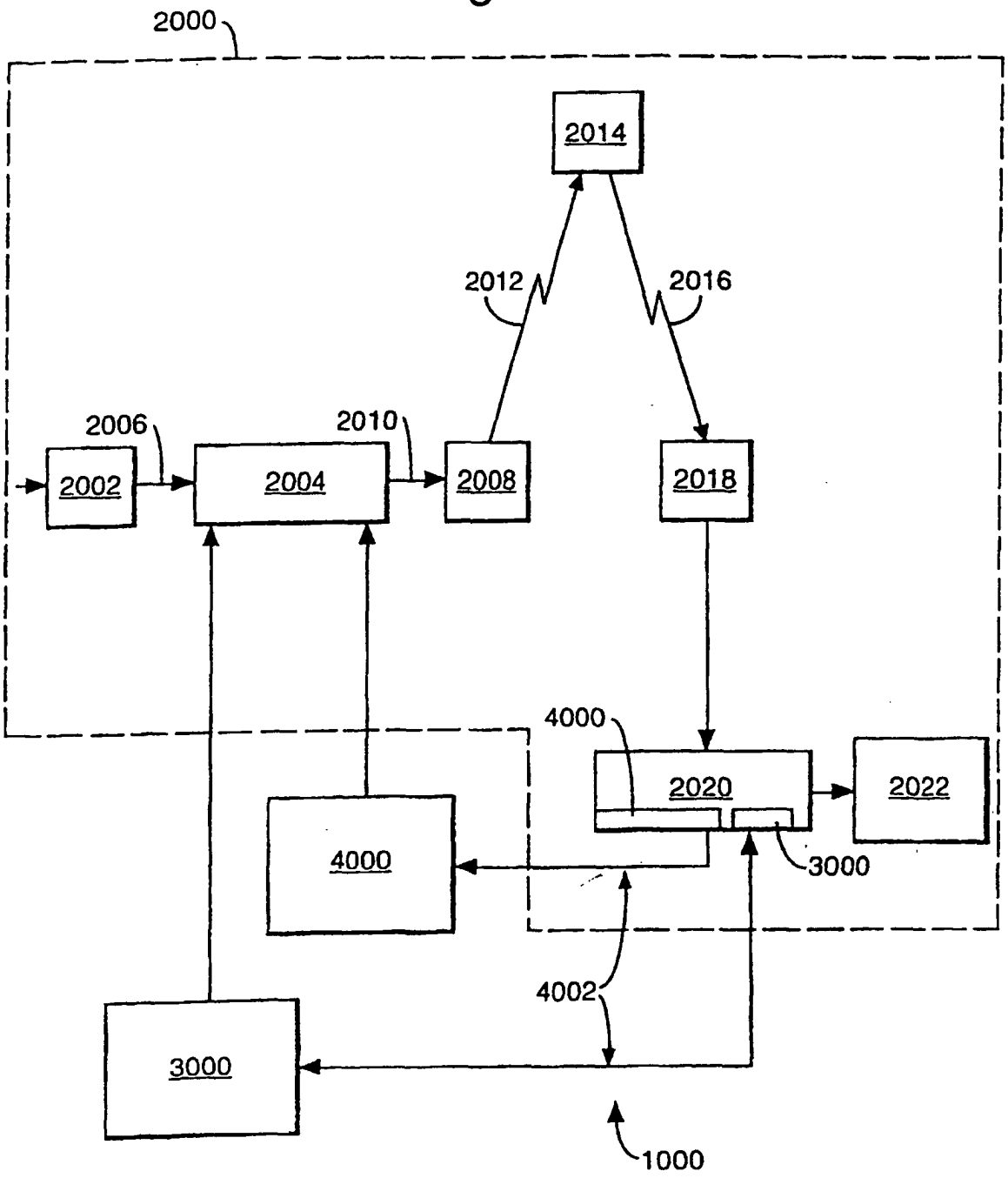


Fig.2.

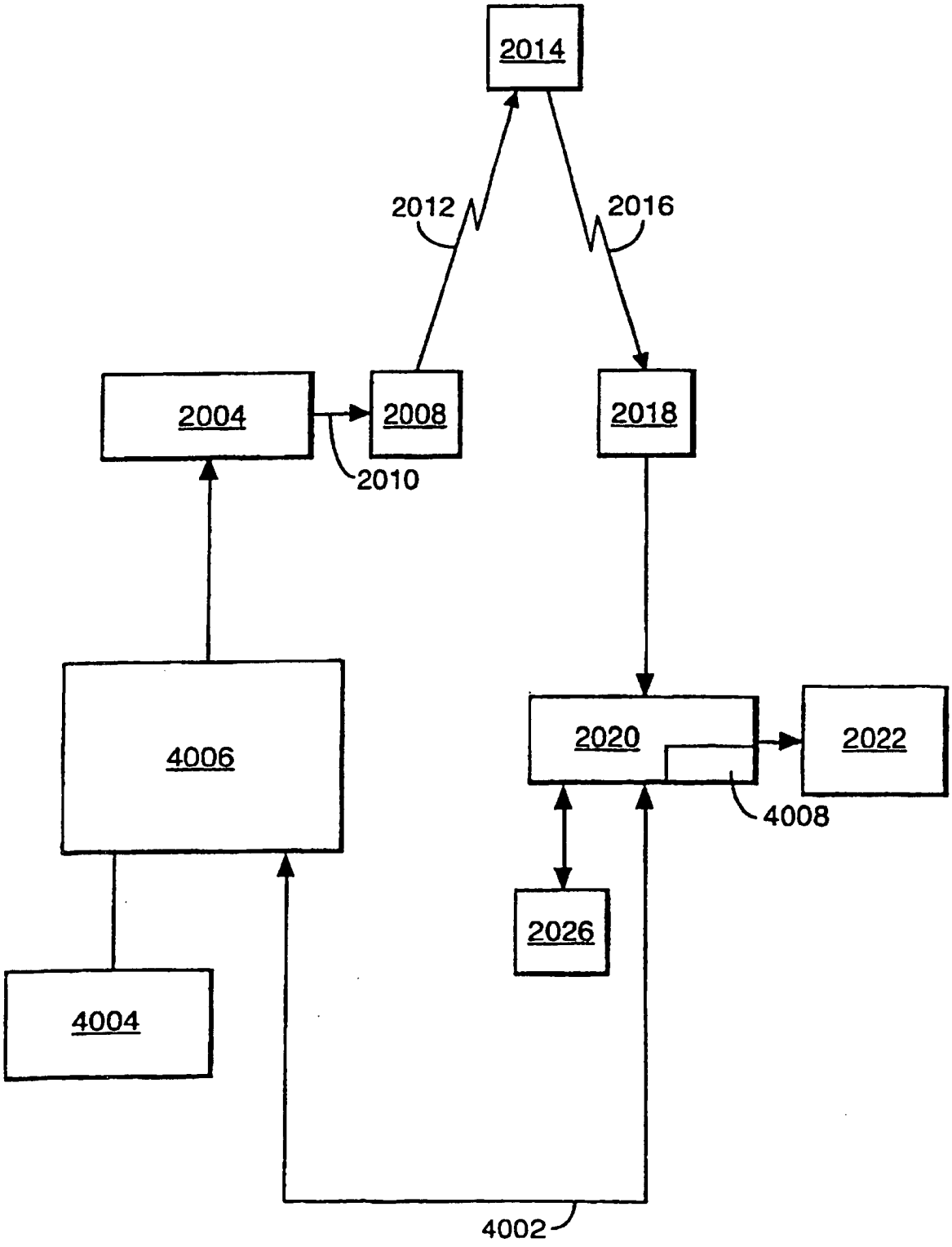

2014
2012
2016
2004
2008
2010
2018
4006
2020
2022
4008
2026
4004
4002

Fig.3.

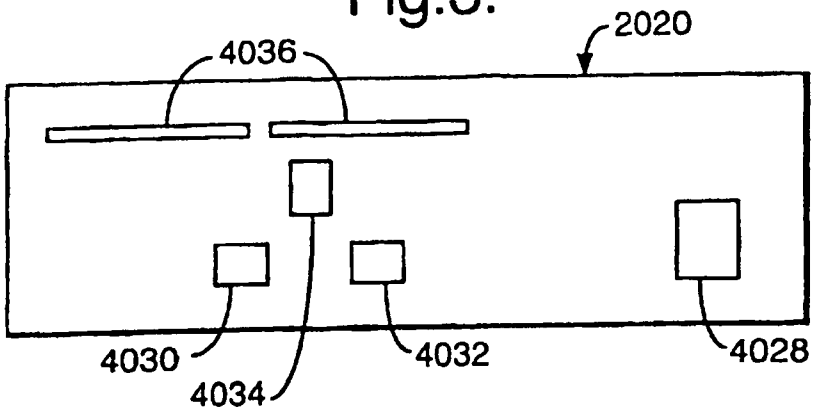

2020
4036
4030
4034
4032
4028

Fig.4.

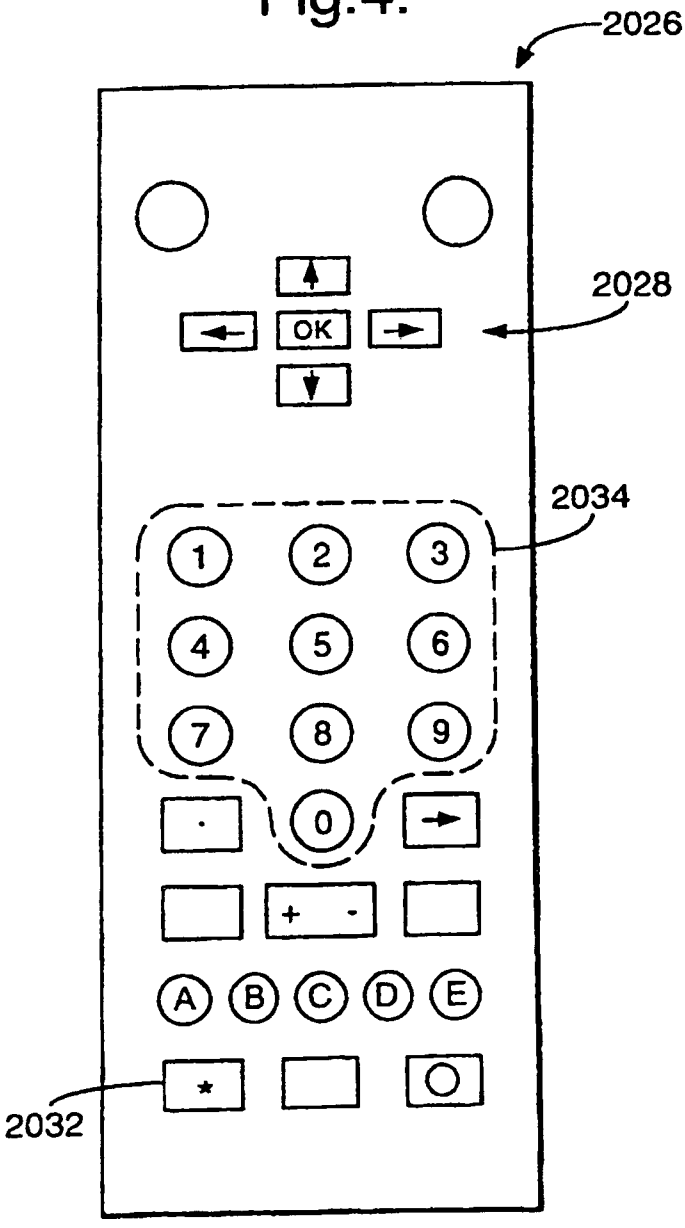

2026
2028
OK
2034
1
2
3
4
5
6
7
8
9
0
+
-
A
B
C
D
E
2032

Fig.5.

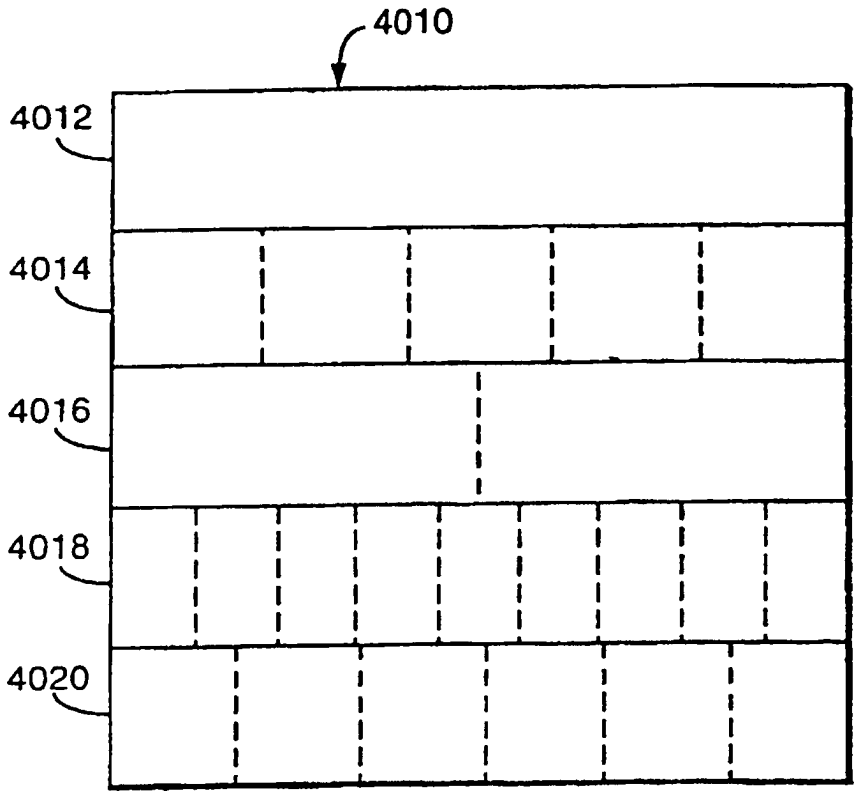


Fig.6.

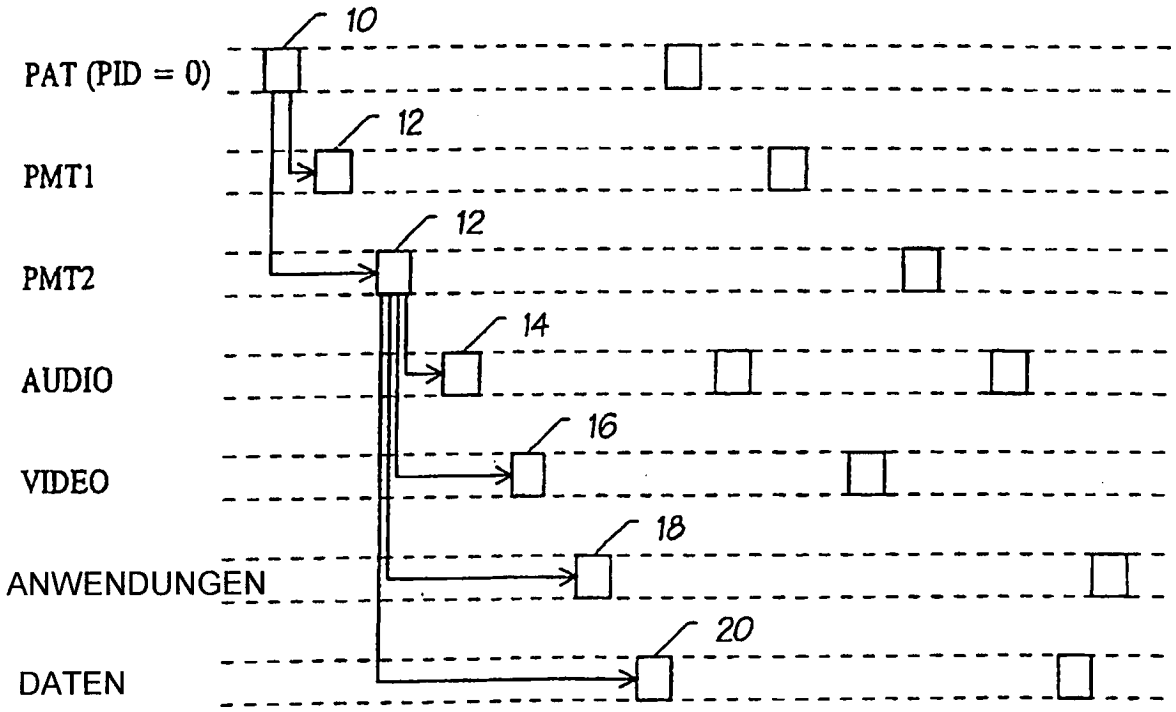

10
PAT (PID = 0)
12
PMT1
12
PMT2
14
AUDIO
16
VIDEO
18
ANWENDUNGEN
20
DATEN

Fig.7.

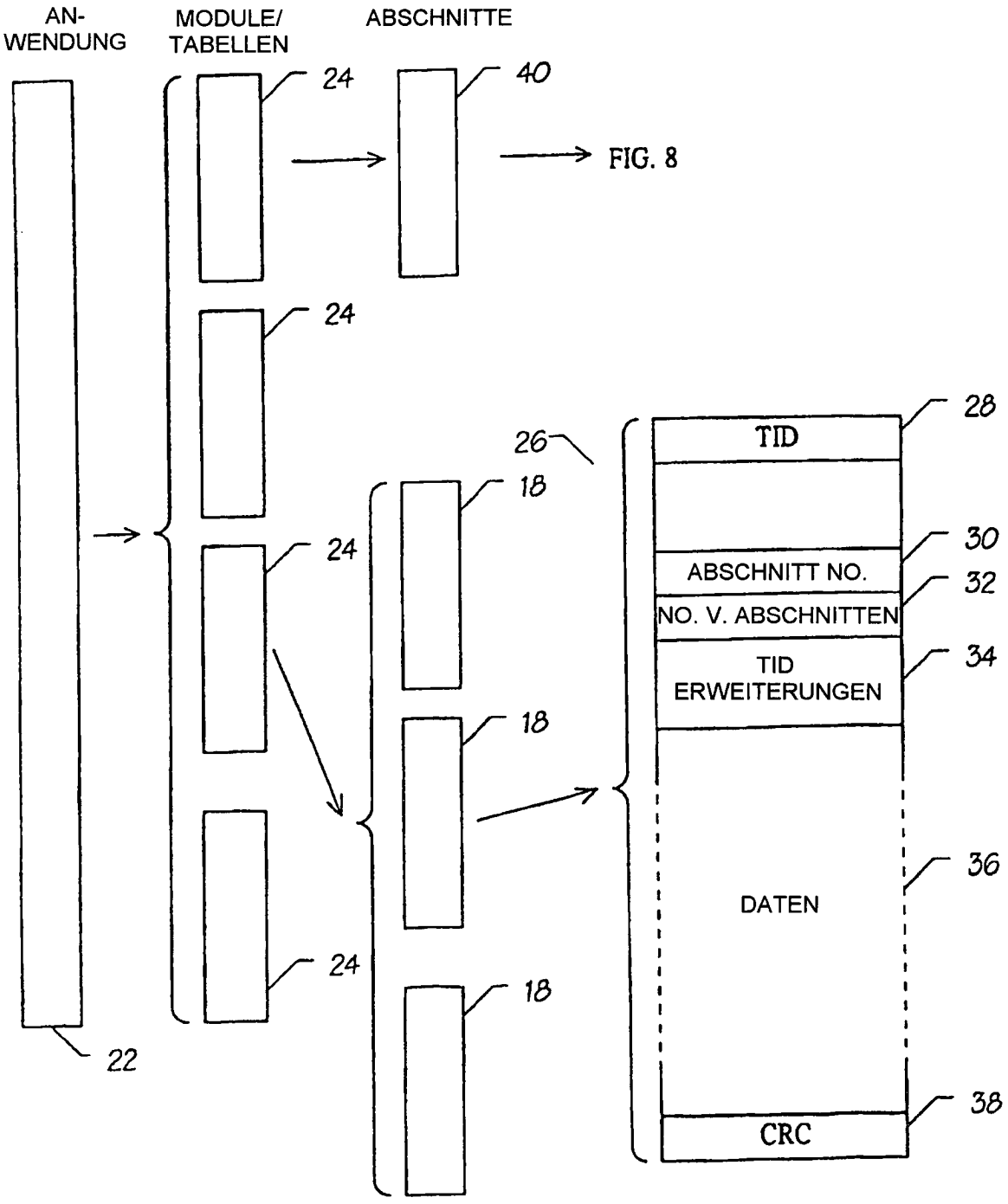

AN-
WENDUNG
MODULE/
TABELLEN
ABSCHNITTE
24
40
FIG. 8
24
24
24
18
18
18
26
28
TID
30
ABSCHNITT NO.
32
NO. V. ABSCHNITTEN
34
TID
ERWEITERUNGEN
36
DATEN
38
CRC
22

Fig.8.

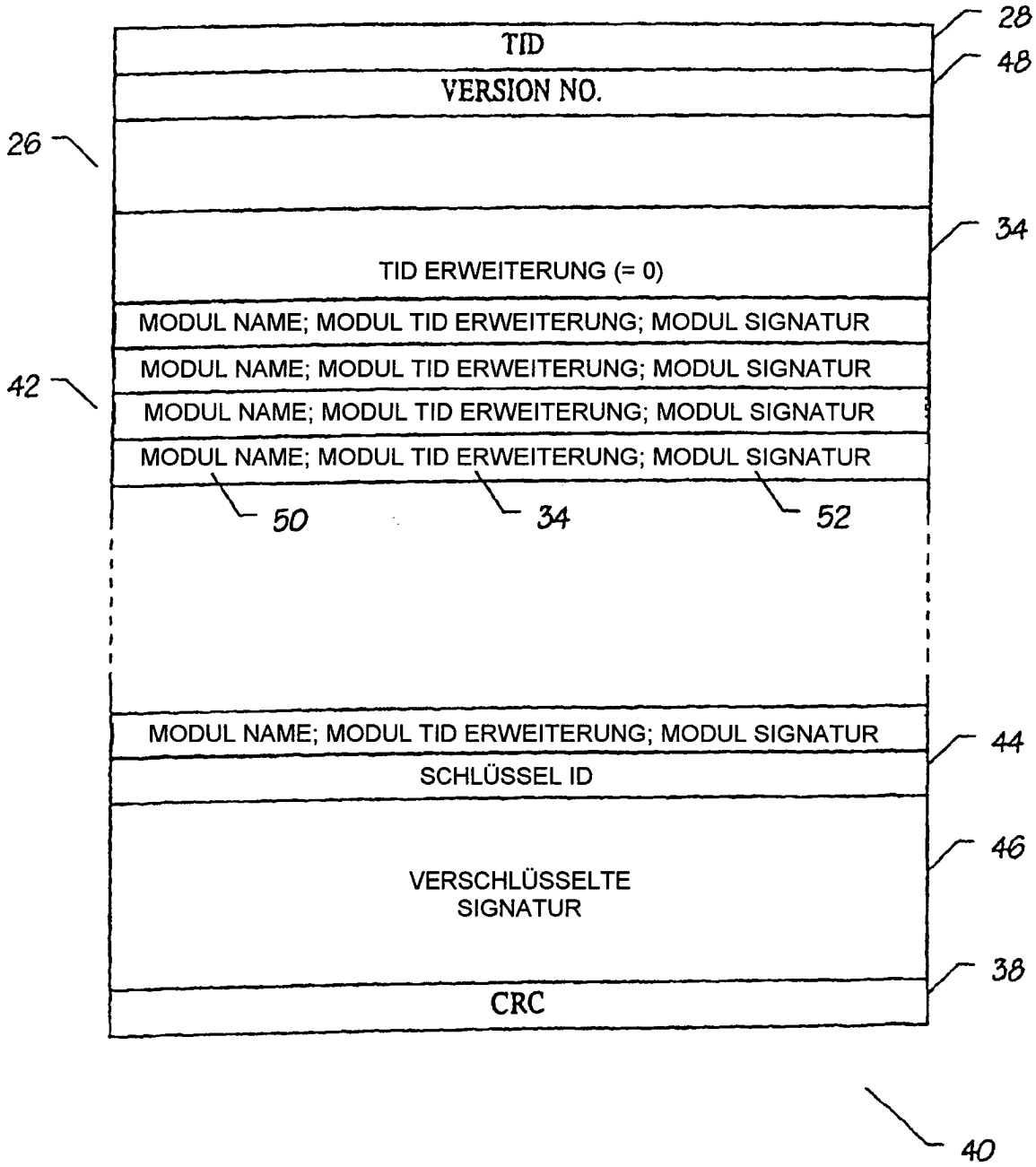

28
TID
48
VERSION NO.
26
34
TID ERWEITERUNG (= 0)
MODUL NAME; MODUL TID ERWEITERUNG; MODUL SIGNATUR
MODUL NAME; MODUL TID ERWEITERUNG; MODUL SIGNATUR
42
MODUL NAME; MODUL TID ERWEITERUNG; MODUL SIGNATUR
MODUL NAME; MODUL TID ERWEITERUNG; MODUL SIGNATUR
50
34
52
MODUL NAME; MODUL TID ERWEITERUNG; MODUL SIGNATUR
44
SCHLÜSSEL ID
46
VERSCHLÜSSELTE SIGNATUR
38
CRC
40

Fig.9.

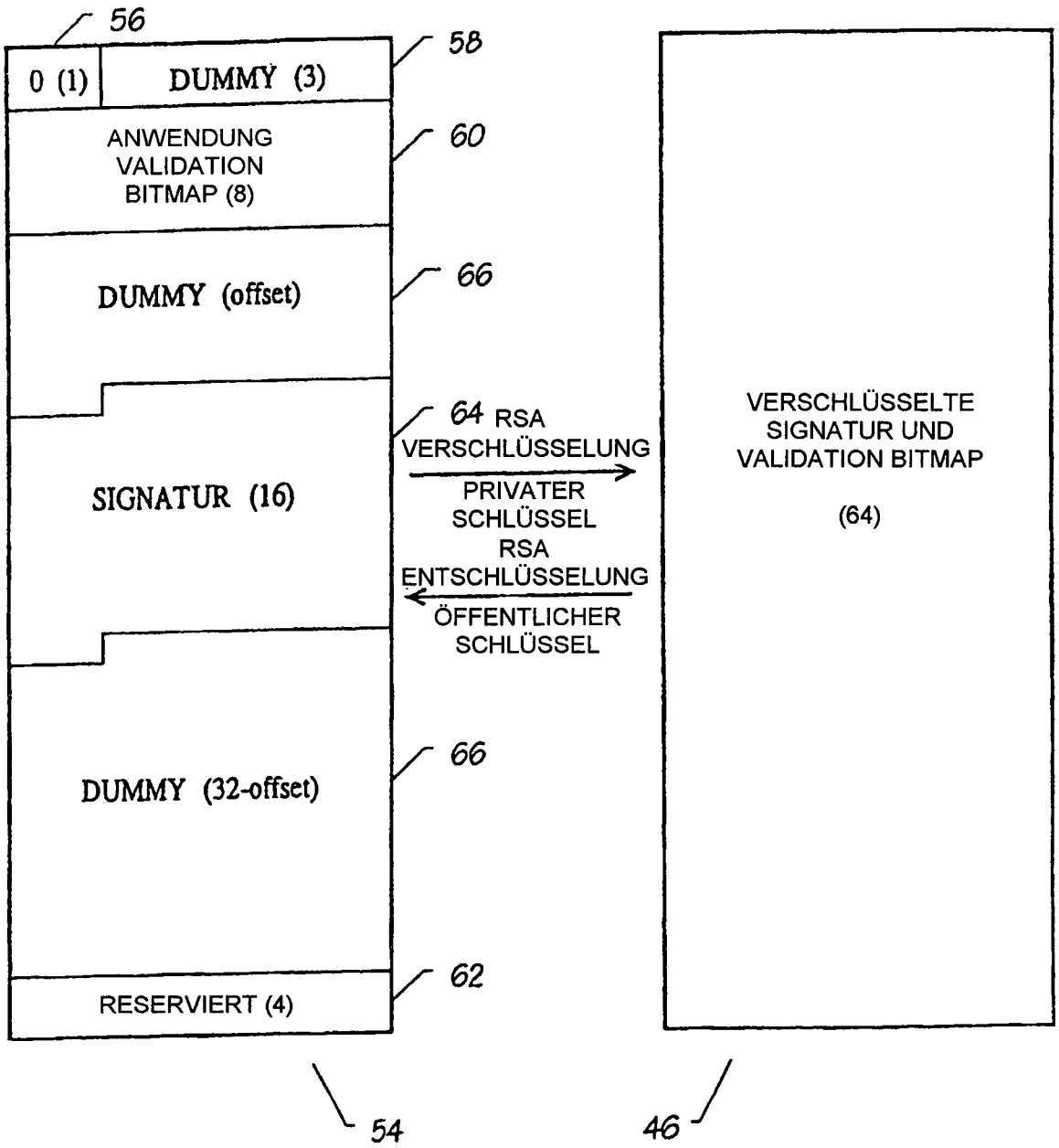
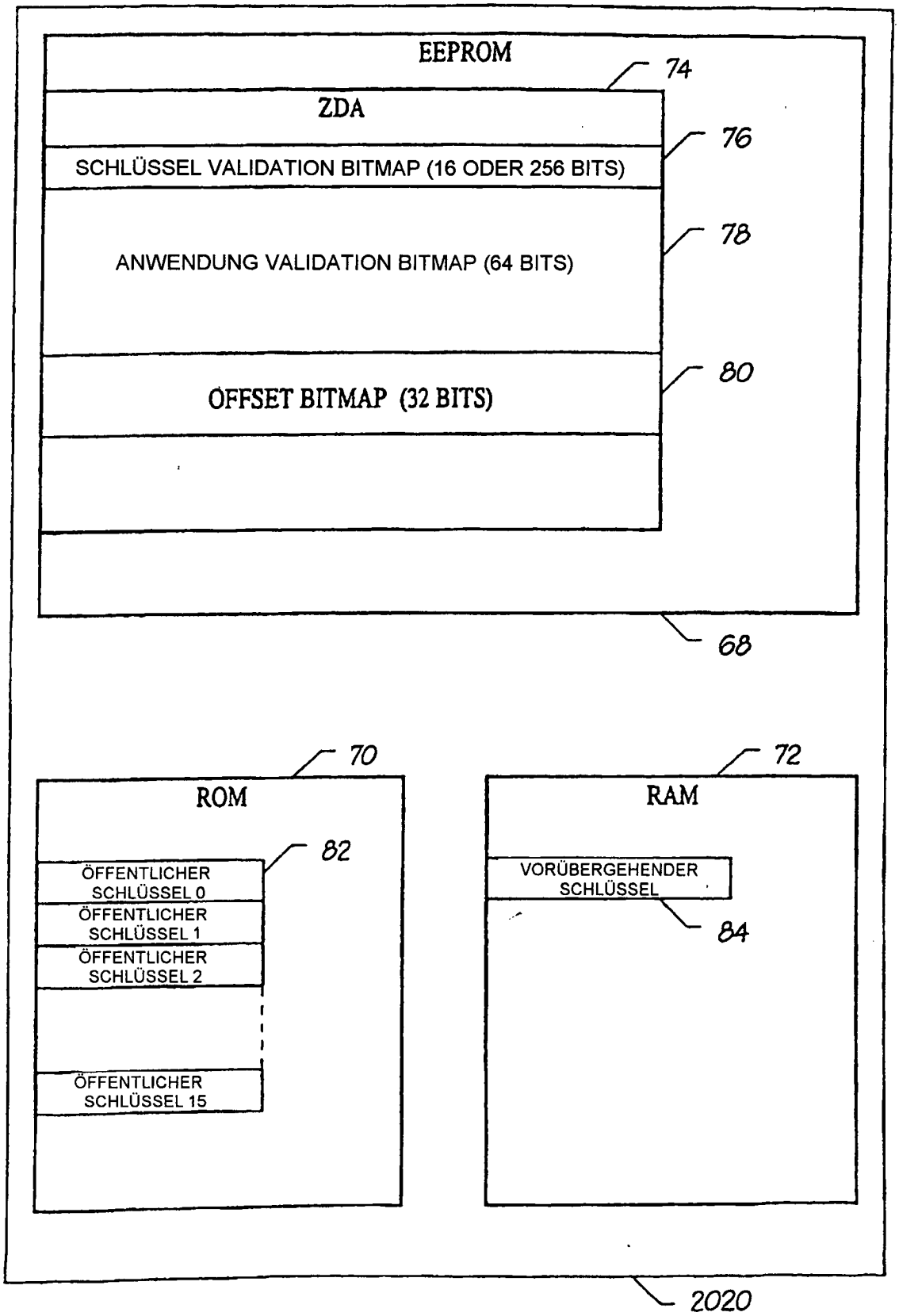

56
58
0 (1)
DUMMY (3)
ANWENDUNG VALIDATION BITMAP (8)
60
DUMMY (offset)
66
64 RSA VERSCHLÜSSELUNG
PRIVATER SCHLÜSSEL
RSA ENTSCHLÜSSELUNG
ÖFFENTLICHER SCHLÜSSEL
SIGNATUR (16)
DUMMY (32-offset)
66
RESERVIERT (4)
62
54
VERSCHLÜSSELTE SIGNATUR UND VALIDATION BITMAP
(64)
46

Fig.10.


EEPROM
74
ZDA
76
SCHLÜSSEL VALIDATION BITMAP (16 ODER 256 BITS)
78
ANWENDUNG VALIDATION BITMAP (64 BITS)
80
OFFSET BITMAP (32 BITS)
68
70
ROM
82
ÖFFENTLICHER SCHLÜSSEL 0
ÖFFENTLICHER SCHLÜSSEL 1
ÖFFENTLICHER SCHLÜSSEL 2
ÖFFENTLICHER SCHLÜSSEL 15
72
RAM
VORÜBERGEHENDER SCHLÜSSEL
84
2020