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

(10) **DE 699 36 157 T2** 2008.01.24

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 646 178 B1**
(21) Deutsches Aktenzeichen: **699 36 157.5**
(96) Europäisches Aktenzeichen: **05 110 160.8**
(96) Europäischer Anmeldetag: **25.03.1999**
(97) Erstveröffentlichung durch das EPA: **12.04.2006**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **23.05.2007**
(47) Veröffentlichungstag im Patentblatt: **24.01.2008**

(51) Int Cl.[8]: ***H04L 9/32*** (2006.01)
H04N 7/167 (2006.01)

(30) Unionspriorität:
98400686 **25.03.1998** **EP**

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

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

(72) Erfinder:
BEUQUE, Jean-Bernard, 92270, Bois Colombes, FR

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

(54) Bezeichnung: **Authentifizierung von in einem digitalen Übertragungssystem übertragenen Daten**

Beschreibung

[0001] Die vorliegende Erfindung betrifft ein Verfahren zum Authentifizieren von in einem digitalen Übertragungssystem gesendeten Daten.

[0002] Die ausgestrahlte Übertragung von digitalen Daten ist auf dem Gebiet der Bezahl-TV-Systeme wohlbekannt, bei denen verwürfelte audiovisuelle Informationen gewöhnlich per Satelliten- oder Satelliten-/Kabelstrecke zu einer Anzahl von Teilnehmern gesendet werden, die jeweils einen Decoder besitzen, der das gesendete Programm zum nachfolgenden Anschauen entwürfeln kann. Es sind terrestrische digitale Ausstrahlungssysteme bekannt. Neuere Systeme verwendeten auch die Ausstrahlungsstrecke zum Senden anderer Daten zusätzlich zu audiovisuellen Daten oder neben diesen, wie zum Beispiel Computerprogramme oder interaktive Anwendungen zu dem Decoder oder zu einem angeschlossenen PC.

[0003] Ein besonderes Problem bei der Übertragung von Anwendungsdaten liegt in der Notwendigkeit, die Integrität und den Ursprung jeglicher solcher Daten zu verifizieren. Da Daten dieser Art zum Umkonfigurieren des Decoders und auch zur Implementierung einer beliebigen Anzahl interaktiver Anwendungen verwendet werden können, ist es entscheidend, daß die empfangenen Daten sowohl vollständig als auch als von einer bekannten Quelle stammend identifiziert sind. Andernfalls kann es zu Betriebsproblemen in Verbindung mit dem Herunterladen unvollständiger Daten kommen, und auch zu dem Risiko, daß der Decoder gegenüber Attacken durch Dritte oder dergleichen offen wird.

[0004] Bisherige Versuche, solche Daten zu authentifizieren, konzentrierten sich auf die Verifikation auf der Ebene der Einkapselung oder Formatierung der Daten in einem Paketstrom. Zum Beispiel beschreibt die europäische Patentanmeldung EP0752786 ein System, bei dem Daten in einer Reihe von Modulen oder bei Verwendung der mit dem MPEG-Standard assoziierten Terminologie, eine Reihe von Tabellen oder Sektionen, eingekapselt werden, wobei die Tabellen oder Sektionen dann in Pakete in einem MPEG-Transportstrom eingekapselt werden.

[0005] Authentifikationsoperationen können in bezug auf die tabulierten Daten ausgeführt werden, wobei eine Verzeichnistabelle zum Beispiel eine Liste aller Tabellen enthält, die Daten für diese Anwendung enthalten, zusammen mit einer Liste von Flash-Werten, die mit jeder Tabelle assoziiert sind, um eine spätere Verifikation von Tabellendaten zu erlauben. Die Verzeichnistabelle kann selbst vor der Übertragung signiert werden, so daß die Informationen in der Verzeichnistabelle und die assoziierten Tabellen nicht modifiziert werden können, ohne die Hash- und Signaturwerte zu ändern.

[0006] Das Problem mit solchen bekannten Systemen liegt in ihrer Ungeeignetheit für die Handhabung von komplexeren Datenorganisationsstrukturen. Insbesondere bedeutet die Verwendung einer einzigen Verzeichnistabelle, die eine vollständige Liste von Hash-Werten für jede assoziierte Tabelle enthält, daß solche Systeme nicht leicht dafür ausgelegt werden können, große oder variabel viele Tabellen handzuhaben.

[0007] Das System ist gleichermaßen schlecht dafür ausgelegt, eine Authentifikation von durch eine Anzahl von Ausstrahlungsbetreibern bereitgestellter Software zu erlauben, da eine einzige MPEG-Verzeichnistabelle alle Tabellen verknüpft und da die Authentifikationsoperationen in einer Phase des Formatierens der Daten in Tabellen zur Paketeinkapselung und -ausstrahlung ausgeführt werden. Diese Operation wird gewöhnlich unter der Kontrolle eines einzigen Betreibers ausgeführt.

[0008] Gemäß einem ersten Aspekt der Erfindung wird ein Verfahren zum Authentifizieren von in einem digitalen Übertragungssystem gesendeten Daten bereitgestellt, gekennzeichnet durch die Organisation der Daten vor der Übertragung zu einer Hierarchie aus mindestens einer Wurzelverzeichniseinheit, einer Unterverzeichniseinheit und einer Dateieinheit, wobei an Daten in einer Datei durch einen Authentifikationsalgorithmus gewirkt wird und ein assoziierter Dateiauthentifikationswert in dem verweisenden Unterverzeichnis gespeichert wird, wobei an diesem Dateiauthentifikationswert seinerseits durch einen Authentifikationsalgorithmus gewirkt und ein assoziierter Unterverzeichnis-Authentifikationswert in dem verweisenden Wurzelverzeichnis gespeichert wird.

[0009] Im Gegensatz zu bekannten Systemen, bei denen eine einzige Tabelle auf alle assoziierten Tabellen verweist, stellt die Verwendung einer mehrfachen Hierarchiestruktur in Verbindung mit der Anwendung eines Authentifikationsalgorithmus in jedem Schritt in der Hierarchie eine sichere und modularisierte Datenstruktur bereit. Da ein Dateiauthentifikationswert in einem Unterverzeichnis seinerseits auf einer höheren Ebene durch einen entsprechenden Wert in dem Wurzelverzeichnis authentifiziert wird, ist es nicht möglich, ein Element auf einer unteren Ebene zu ändern, ohne die authentifizierenden Werte auf einer höheren Ebene zu ändern (und umgekehrt).

[0010] Vorzugsweise wird die Authentifikation der Dateidaten durch Anwenden eines Hash-Algorithmus auf einen Teil oder alle der Dateidaten ausgeführt, wobei der resultierende Hash-Wert als der Dateiauthentifikationswert in dem verweisenden Unterverzeichnis gespeichert wird. Gleichermaßen kann eine

Authentifikation eines Unterverzeichnisses ausgeführt werden, indem man einen Hash-Algorithmus auf den Dateiauthentifikationswert (und gegebenenfalls andere Daten) anwendet, wobei der resultierende Hash-Wert als der Unterverzeichnis-Authentifikationswert in dem verweisenden Wurzelverzeichnis gespeichert wird.

[0011] Es können andere Ausführungsformen in Betracht gezogen werden, wenn zum Beispiel Dateidaten gemäß einem Verschlüsselungsalgorithmus verschlüsselt werden und der Verschlüsselungsschlüssel (oder seine identifizierende Schlüsselnummer) als der in dem Unterverzeichnis gespeicherte Authentifikationswert verwendet wird. Dieser Dateischlüssel kann seinerseits verschlüsselt werden und der Verschlüsselungsschlüssel wird als der Authentifikationswert in dem Wurzelverzeichnis gespeichert usw. Obwohl sie möglich ist, ist diese Ausführungsform aufgrund der vergrößerten Komplexität der zur Erzeugung von Verschlüsselungsschlüsselwerten notwendigen Operationen komplizierter zu implementieren.

[0012] Im Gegensatz dazu ermöglicht die Verwendung des Hash-Algorithmus zum Ausführen der Authentifikation jedes Moduls die Ausführung einer besonders einfachen und schnellen Prüfung der Integrität jedes Moduls. Bei einer Ausführungsform kann man einen einfachen Hash-Algorithmus, wie zum Beispiel eine Prüfsummenberechnung, verwenden. Dies würde jedoch keine Detektion einer Falsifizierung ermöglichen, da es relativ einfach ist, zu bestimmen, wie sich etwaige Änderungen in einer Nachricht auf den Hash-Wert auswirken.

[0013] Vorzugsweise entspricht der Hash-Algorithmus einem kryptographisch sicheren Algorithmus, der aus einer gegebenen Menge von Daten einen im wesentlichen eindeutigen Flash-Wert (15) erzeugt. Es können geeignete Hash-Algorithmen für diesen Zweck verwendet werden, darunter zum Beispiel der MD5-Algorithmus (Message Digest version 5) oder der SHA (Secure Hash Algorithm).

[0014] Vorteilhafterweise wird die Authentifikation von Dateidaten für mehrere Dateien durch Anwenden eines Hash-Algorithmus auf eine Akkumulation von Daten aus mehreren Dateien angewandt, um einen einzigen Hash-Wert zu erzeugen. Gleichermaßen kann die Authentifikation einer Anzahl von Unterverzeichnissen ausgeführt werden, indem man einen Hash-Algorithmus auf eine Akkumulation von Dateiauthentifikationswerten aus mehreren Unterverzeichnissen (und gegebenenfalls anderen Daten) anwendet, um einen einzigen Hash-Wert zu erzeugen.

[0015] Die Verwendung eines kumulativen Hash-Prozesses zur Abdeckung mehrerer Datenmodule (Dateien, Unterverzeichnisse usw.) auf einer niedrigeren Schicht vereinfacht das System zum Beispiel im Vergleich mit Systemen, die Listen individueller Hash-Werte für jedes Modul speichern, weiter. Dadurch kann das System wiederum die aus jeder Ebene notwendigen Berechnungsschritte reduzieren und reduziert die Größe von in einer höheren Schicht gespeicherten Authentifikationsdaten.

[0016] Im Fall der Ausführungsformen, die einen Hash-Algorithmus zum Authentifizieren jeder Schicht verwenden, ist das System "offen", das heißt, alle Hash-Werte sind bis zum Wurzelverzeichnis herauf lesbar. Da Hash-Algorithmen öffentlich verfügbar sind, könnten Dritte theoretisch gespeicherte Daten z.B. auf Dateiebene ohne Detektion ändern, wenn auch die entsprechenden Hash-Werte auf der Unterverzeichnis- und Wurzelverzeichnisebene gleichzeitig geändert würden.

[0017] Um dies zu vermeiden, wird an mindestens einem Teil der in dem Wurzelverzeichnis gespeicherten Daten durch einen Geheimschlüssel eines Verschlüsselungsalgorithmus gewirkt und der resultierende verschlüsselte Wert in dem Wurzelverzeichnis gespeichert. Vorzugsweise entspricht der verschlüsselte Wert einer digitalen Signatur. Zu geeigneten Algorithmen mit privatem/öffentlichem Schlüssel für diesen Zweck gehört zum Beispiel der RSA-Algorithmus.

[0018] Vorteilhafterweise umfassen die durch den Geheimschlüssel zur Erzeugung einer in dem Wurzelverzeichnis gespeicherten Signatur verschlüsselten Daten mindestens einen oder mehrere Unterverzeichnis-Authentifikationswerte. Dessen ungeachtet ist es möglich, in Betracht zu ziehen, daß andere Daten in dem Wurzelverzeichnis als die Unterverzeichnis-Authentifikationswerte signiert werden, um das System zu "schließen".

[0019] Bei einer Alternative zur Erzeugung einer Signatur kann das Wurzelverzeichnis ganz oder teilweise verschlüsselt oder verwürfelt werden, wobei der Empfänger einen äquivalenten Schlüssel zum Entschlüsseln der verschlüsselten Wurzelverzeichnisdaten besitzt. In diesem Fall kann man einen symmetrischen Schlüsselalgorithmus wie etwa DES verwenden.

[0020] Obwohl der Authentifikationsprozeß oben mit Bezug auf zwei hierarchische Ebenen beschrieben wurde, versteht sich, daß man ähnliche Authentifikationsschritte ad infinitum für weitere Dateien, Unterverzeichnisse, Wurzelverzeichnisse usw., auf die verwiesen wird, ausführen kann.

[0021] Obwohl die Struktur der Klarheit der Sprache halber als Wurzelverzeichnis/Unterverzeichnis/Datei definiert wurde, wird ähnlich keine konkrete Eigenschaft jeder Einheit in einer Schicht angenommen,

mit Ausnahme des Verweises auf eine Einheit niedrigerer Schichten durch zwei Einheiten höherer Schichten. Es versteht sich, daß die Datenstruktur gleichermaßen ein Wurzelverzeichnis/Unterverzeichnis/zweites Wurzelverzeichnis oder jede beliebige andere Kombination sein kann.

[0022] Die folgenden beschriebenen Ausführungsformen konzentrieren sich auf eine Einheit in einer niedrigeren Schicht, d.h. auf die durch ein Verzeichnis oder Unterverzeichnis verwiesen wird. Wie deutlich werden wird, kann diese Einheit, obwohl auf sie von einer höheren Schicht verwiesen wird, ungeachtet selbst eine Verzeichniseinheit, Unterverzeichniseinheit usw. sein.

[0023] Bei einer Ausführungsform umfaßt eine Einheit, auf die verwiesen wird, einen durch einen Geheimschlüssel erzeugten verschlüsselten Wert, wobei ein Authentifikationswert für diese Einheit auf der Basis der Ergebnisse eines Authentifikationsalgorithmus an dem verschlüsselten Wert berechnet und in der verweisenden Einheit gespeichert wird. Wie bei der oben beschriebenen Äquivalent-Wurzelverzeichnisausführungsform kann insbesondere eine verwiesene Einheit signiert werden, wobei der Authentifikationswert für diese Einheit als Ergebnis einer Hash-Funktion an dieser Signatur berechnet wird.

[0024] Die verwiesene Einheit kann zum Beispiel einer Datei oder einem Unterverzeichnis entsprechen. Diese Ausführungsform ist jedoch insbesondere für die Situation ausgelegt, in der die verwiesene Einheit ein Wurzelverzeichnis für eine weitere Menge von Daten ist, z.B. Daten eines anderen Ursprungs, und bei dem die verwiesene Wurzeleinheit auch eine Signatur enthält. In diesem Fall kann ein erster Betreiber Daten bis zu der Ebene des Wurzelverzeichnisses herauf assemblieren und signieren.

[0025] Danach kann ein zweiter Betreiber auf diese Daten verweisen, ohne den Verschlüsselungsschlüssel zu kennen, wobei eine etwaige Strecke einfach in der verweisenden Einheit durch den Hash-Wert der Signatur in dem verwiesenen Wurzelverzeichnis authentifiziert wird. Die Authentifikation beider Mengen von Daten wird natürlich nur einem Empfänger möglich sein, der die notwendigen Schlüssel zum Verifizieren der Signaturen in beiden Wurzelverzeichnissen besitzt.

[0026] Wie oben beschrieben, kann die vorliegende Erfindung auf jede beliebige Menge von Mehrfach-Hierarchiedateneinheiten angewandt werden. Sie kann sogar auf die Organisation von Tabellen oder Paketen in einem Transportstrom angewandt werden, wenn mehrere Ebenen eines Wurzelverzeichnisses, Unterverzeichnisses, einer Datei usw. in einem Paketstrom bereitgestellt werden können. Die vorliegende Erfindung ist jedoch insbesondere auf den Fall anwendbar, bei dem die Einheiten einer Menge von Daten-Dateien entsprechen, die in Datentabellen oder -sektionen eingekapselt sind, wobei diese Tabellen danach in Datenpakete eingekapselt werden, um einen Transportstrom zu bilden.

[0027] Im Gegensatz zur Authentifikation auf Paket- oder Tabellenebene ermöglicht diese Ausführungsform vollständige Unabhängigkeit zwischen dem Assemblieren authentifizierter Daten und ihrer Einkapselung in einen Transportstrom und erleichtert wieder die Zuführung von Software von verschiedenen Quellen in dem durch einen einzigen Ausstrahlungsbetreiber kontrollierten Transportstrom. Gemäß dieser Ausführungsform authentifizierte Daten können sogar über verschiedene Übertragungsrouten (z.B. eine bidirektionale Telecom-Strecke oder eine Satellitenstrecke) gesendet werden, wobei alternative Einkapselungsformate zum Senden der Daten benutzt werden.

[0028] Wie bereits erwähnt, hat die Verwendung eines Authentifikationsprozesses, der vor der Vorbereitung von Daten für die Übertragung angewandt wird, den Effekt, daß die Daten danach durch eine beliebige Anzahl von Kanälen, wie zum Beispiel einen Ausstrahlungskanal oder einen Telecom-Kanal, zu einem Empfänger geroutet werden können, ohne den Authentifikationsprozeß zu ändern. Sobald ein Empfänger oder Decoder die Daten-Dateien aus dem mit der Übertragungsroute assoziierten Format rekonstituiert hat, kann gleichermaßen eine Verifikation unabhängig von dem gewählten Übertragungsmodus an diesen Daten ausgeführt werden.

[0029] Jegliche oder alle Merkmale des ersten Aspekts der Erfindung und ihre bevorzugten Ausführungsformen können natürlich mit dem zweiten und dritten Aspekt der Erfindung kombiniert werden.

[0030] Die vorliegende Erfindung wurde oben in bezug auf die Schritte zum Erzeugen von Authentifikationsdaten vor der Übertragung beschrieben. Die Erfindung gilt in ihren allgemeinsten und bevorzugten Ausführungsformen gleichermaßen für die in einem Empfänger zum Verifizieren dieser Daten ausgeführten umgekehrten Schritte.

[0031] In ihren allgemeinsten Aspekten kann die vorliegende Erfindung auf jedes digitale Übertragungssystem angewandt werden. Die Erfindung wird jedoch vorzugsweise auf ein digitales Fernsehsystem angewandt und insbesondere auf Datenmodule, die Anwendungssoftware zur Verwendung in einem Empfänger/Decoder des digitalen Fernsehsystems führen.

[0032] Der Begriff "digitales Übertragungssystem" umfaßt im vorliegenden Kontext jedes beliebige Übertragungssystem zum Senden oder Ausstrahlen

zum Beispiel von hauptsächlich audiovisuellen oder Multimedia-Digitaldaten. Obwohl die vorliegende Erfindung besonders auf ein digitales Fernsehausstrahlungssystem anwendbar ist, kann die Erfindung auch auf ein festes Telekommunikationsnetz für Multimedia-Internet-Anwendungen, auf ein Überwachungskamerasystem usw. angewandt werden. Es versteht sich, daß der Begriff "digitales Fernsehsystem" zum Beispiel jedes Satelliten-, terrestrische, Kabel- und anderweitige System umfaßt.

[0033] Der Begriff "Empfänger/Decoder" oder "Decoder", der in der vorliegenden Anmeldung benutzt wird, kann einen Empfänger zum Empfangen entweder von codierten oder von nichtcodierten Signalen, wie zum Beispiel Fernseh- und/oder Rundfunksignalen bedeuten, die ausgestrahlt oder durch bestimmte andere Mittel gesendet werden können. Der Begriff kann auch einen Decoder zum decodieren von Empfangsignalen bedeuten. Ausführungsformen von solchen Empfängern/Decodern können einen mit dem Empfänger integralen Decoder zum Decodieren der Empfangssignale zum Beispiel in einem "Digitalreceiver" umfassen, wobei ein solcher Decoder in Kombination mit einem physisch getrennten Empfänger fungiert oder ein solcher Decoder zusätzliche Funktionen enthält, wie zum Beispiel einen Web-Browser und/oder mit anderen Einrichtungen wie zum Beispiel einem Videorecoder oder einem Fernsehapparat integriert ist.

[0034] Der Begriff MPEG bezieht sich auf die Datenübertragungsnormen, die von der internationalen Normenorganisationsarbeitsgruppe "Motion Pictures Expert Group" entwickelt werden, und insbesondere, aber nicht ausschließlich auf die MPEG-2-Norm, die für digitale Fernsehanwendungen entwickelt wird und in den Schriften ISO 13818-1, ISO 13818-2, ISO 13818-3 und ISO 13818-4 dargelegt wird. Im Kontext der vorliegenden Patentanmeldung umfaßt der Begriff MPEG alle Varianten und Modifikationen von auf das Gebiet der digitalen Datenübertragung anwendbaren MPEG-Formaten.

[0035] Der Begriff DSMCC bezieht sich auf die in den MPEG-Schriften und in der aktuellen Schrift ISO 13818-6 beschriebenen Daten-Dateiformat-Normen.

[0036] Es wird nun lediglich als Beispiel eine bevorzugte Ausführungsform der Erfindung mit Bezug auf die beigefügten Figuren beschrieben. Es zeigen:

[0037] Fig. 1 eine schematische Skizze eines digitalen Fernsehsystems zur Verwendung mit der vorliegenden Erfindung;

[0038] Fig. 2 die Struktur eines Decoders des Systems von Fig. 1;

[0039] Fig. 3 die Struktur einer Anzahl von Komponenten in dem MPEG-Ausstrahlungs-Transportstrom;

[0040] Fig. 4 die Unterteilung einer Softwareanwendung in einer Anzahl von MPEG-Tabellen;

[0041] Fig. 5 die Beziehung zwischen DSMCC-Daten-Dateien und den letztendlich produzierten MPEG-Tabellen;

[0042] Fig. 6 die im Kontext von DSMCC definierte Client-, Server-, Netzwerkmanager-Beziehung;

[0043] Fig. 7 die authentifizierten Verzeichnis-, Unterverzeichnis- und Dateiobjekte bei dieser Ausführungsform der Erfindung.

[0044] Fig. 1 zeigt eine Übersicht über ein digitales Fernsehsystem **1**. Die Ausführungsform enthält ein zum größten Teil herkömmliches digitales Fernsehsystem **2**, das das bekannte MPEG-2-Kompressionssystem zum Senden komprimierter digitaler Signale verwendet. Genauer gesagt empfängt der MPEG-2-Kompressor **3** in einer Ausstrahlungszentrale einen digitalen Signalstrom (typischerweise einen Strom von Videosignalen). Der Kompressor **3** ist durch die Verknüpfung **5** mit einem Multiplexer und Verwürfler **4** verbunden.

[0045] Der Multiplexer **4** empfängt mehrere weitere Eingangssignale, assembliert den Transportstrom und sendet komprimierte digitale Signale über die Verknüpfung **7**, die natürlich vielfältige Formen annehmen kann, darunter Telekommunikationsstrecken, zu einem Sender **6** der Ausstrahlungszentrale. Der Sender **6** sendet elektromagnetische Signale über die Aufwärtsstrecke **8** in Richtung eines Satellitentransponders **9**, in dem sie elektronisch verarbeitet und über die nationale Abwärtsstrecke **10** gewöhnlich über eine Schüssel, die der Endbenutzer besitzt oder mietet, zu dem Bodenempfänger **12** ausgestrahlt werden. Die von dem Empfänger **12** empfangenen Signale werden zu einem integrierten Empfänger/Decoder **13** gesendet, den der Endbenutzer besitzt oder mietet, und mit dem Fernsehgerät **14** des Endbenutzers verbunden. Der Empfänger/Decoder **13** decodiert das komprimierte MPEG-2-Signal zu einem Fernsehsignal für das Fernsehgerät **14**.

[0046] Es sind natürlich andere Transportkanäle für die Übertragung der Daten möglich, wie etwa terrestrische Ausstrahlung, Kabelübertragung, kombinierte Satelliten/Kabelstrecken, Telefonnetze usw.

[0047] In einem mehrkanaligen System behandelt der Multiplexer **4** von einer Anzahl paralleler Quellen empfangene Audio- und Videoinformationen und tritt mit dem Sender **6** in Dialog, um die Informationen über eine entsprechende Anzahl von Kanälen auszustrahlen. Zusätzlich zu audiovisuellen Informationen

können in bestimmte oder alle dieser Kanäle mit den gesendeten digitalen Audio- und Videoinformationen verschachtelt Nachrichten oder Anwendungen oder eine beliebige andere Art von digitalen Daten eingeführt werden. In einem solchen Fall wird ein Strom digitaler Daten zum Beispiel in Form von Software-Dateien und Nachrichten des Formats DSM-CC durch den Kompressor **3** komprimiert und zu dem MPEG-Format paketiert. Das Herunterladen von Softwaremodulen wird später ausführlicher beschrieben.

[0048] Ein System **15** für bedingten Zugang ist mit dem Multiplexer **4** und dem Empfänger/Decoder **13** verbunden und befindet sich teilweise in der Ausstrahlungszentrale und teilweise in dem Decoder. Es ermöglicht dem Endbenutzer den Zugang zu digitalen Fernsehausstrahlungen von einem oder mehreren Ausstrahlungsanbietern. Eine Chipkarte mit der Fähigkeit zum Dechiffrieren von Nachrichten in bezug auf kommerzielle Angebote (d.h. eines oder mehrere von dem Ausstrahlungsanbieter vertriebene Fernsehprogramme) kann in den Empfänger/Decoder **13** eingeführt werden. Unter Verwendung des Decoders **13** und der Chipkarte kann der Endbenutzer kommerzielle Angebote entweder im Abonnementmodus oder in einem Pay-Per-View-Modus kaufen. In der Praxis kann der Decoder dafür konfiguriert sein, Mehrfachzugangs-Steuersysteme z.B. des Typs Simulcrypt oder Multicrypt zu behandeln.

[0049] Wie bereits erwähnt, werden durch das System gesendete Programme in dem Multiplexer **4** verwürfelt, wobei die auf eine gegebene Übertragung angewandten Bedingungen und Verschlüsselungsschlüssel durch das Zugangskontrollsystem **15** bestimmt werden. Die Übertragung verwürfelter Daten auf diese Weise ist auf dem Gebiet der Pay-TV-Systeme wohlbekannt. Typischerweise werden verwürfelte Daten zusammen mit einem Steuerwort zum Entwürfeln der Daten gesendet, wobei das Steuerwort selbst durch einen sogenannten Ausnutzungsschlüssel verschlüsselt und in verschlüsselter Form gesendet wird.

[0050] Die verwürfelten Daten und das verschlüsselte Steuerwort werden dann durch den Decoder **13** empfangen, der Zugang zu einem äquivalent des Ausnutzungsschlüssels hat, das auf einer in dem Decoder eingeführten Chipkarte gespeichert ist, um das verschlüsselte Steuerwort zu entschlüsseln und danach die gesendeten Daten zu entwürfeln. Zum Beispiel empfängt ein Teilnehmer, der bezahlt hat, in einer ausgestrahlten monatlichen EMM (Berechtigungsverwaltungsnachricht) den notwendigen Ausnutzungsschlüssel zum Entschlüsseln des verschlüsselten Steuerworts, um so ein Anschauen der Übertragung zu erlauben. Zusätzlich zu ihrer Verwendung beim Entschlüsseln von audiovisuellen Fernsehprogrammen können ähnliche Ausnutzungsschlüssel zur Verwendung bei der Verifikation von anderen Daten, wie zum Beispiel Softwaremodulen, erzeugt und gesendet werden, wie später beschrieben werden wird.

[0051] Ein interaktives System **16**, das auch mit dem Multiplexer **4** und dem Empfänger/Decoder **13** verbunden ist und sich wiederum teilweise in der Ausstrahlungszentrale und teilweise in dem Decoder befindet, ermöglicht dem Endbenutzer einen Dialog mit verschiedenen Anwendungen über einen Modem-Rückkanal **17**. Der Modem-Rückkanal kann auch für in dem System **15** für bedingten Zugang verwendete Übermittlungen verwendet werden. Ein interaktives System kann zum Beispiel dazu verwendet werden, es dem Zuschauer zu ermöglichen, unmittelbar mit der Übertragungszentrale zu kommunizieren, um die Authorisation zum Anschauen eines bestimmten Ereignisses, Herunterladen einer Anwendung usw. zu bestellen.

[0052] Mit Bezug auf Fig. 2 werden nun die physischen Elemente des Empfängers/Decoders **13** oder Digitalreceivers, die für die Verwendung in der vorliegenden Erfindung ausgelegt sind, kurz beschrieben. Die in dieser Figur gezeigten Elemente werden über Funktionsblöcke beschrieben.

[0053] Der Decoder **13** umfaßt einen Zentralprozessor **20** mit assoziierten Speicherelementen, der dafür ausgelegt ist, Eingangsdaten von einer seriellen Schnittstelle **21**, einer parallelen Schnittstelle **22** und einem Modem **23** (mit dem Modem-Rückkanal **17** von Fig. 1 verbunden) zu empfangen.

[0054] Zusätzlich ist der Decoder dafür ausgelegt, Eingangssignale von einer Infrarotfernbedienung **25** über eine Steuereinheit **26** und von Schalterkontakten **24** auf der Vorderseite des Decoders zu empfangen. Außerdem besitzt der Decoder zwei Chipkartenleser **27**, **28**, die dafür ausgelegt sind, Bank- bzw. Abonnement-Chipkarten **29**, **30** zu lesen. Eingaben können auch über eine (nicht gezeigte) Infrarottastatur empfangen werden. Der Abonnement-Chipkartenleser **28** tritt mit einer eingefügten Abonnementkarte und mit einer Einheit **29** für bedingten Zugang in Eingriff, um das notwendige Steuerwort einem Demultiplexer/Entwürfler **30** zuzuführen, um die Entwürflung des verschlüsselten ausgestrahlten Signals zu ermöglichen. Der Decoder enthält außerdem einen herkömmlichen Tuner **31** und einen Demodulator **32** zum Empfangen und Demodulieren der Satellitenübertragung vor dem Filtern und Demultiplexen durch die Einheit **30**.

[0055] Die Verarbeitung von Daten in dem Decoder wird im allgemeinen durch den Zentralprozessor **20** abgewickelt. Die Architektur der Software **15** des Zentralprozessors entspricht einer virtuellen Maschine, die mit einem Betriebssystem einer niedrigeren

Ebene, das in Hardwarekomponenten des Decoders implementiert ist, in Dialog tritt.

[0056] Es wird nun mit Bezug auf Fig. 3 und Fig. 4 die Paketstruktur von Daten in dem von dem Sender zu dem Decoder gesendeten ausgestrahlten MPEG-Transportstrom beschrieben. Es versteht sich, daß, obwohl sich die Beschreibung auf das in der MPEG-Norm verwendete Tabulationsformat konzentriert, dieselben Prinzipien gleichermaßen für andere paketierte Datenstromformate gelten.

[0057] Inbesondere mit Bezug auf Fig. 3 enthält ein MPEG-Bitstrom eine Programmzugangstabelle ("PAT") **40** mit einer Paketidentifikation ("PID") von 0. Die PAT enthält Verweise auf die PIDs der Programmabbildungstabellen ("PMTs") **41** einer Anzahl von Programmen. Jede PMT enthält einen Verweis auf die PIDs der Ströme der Audio-MPEG-Tabellen **42** und Video-MPEG-Tabellen **43** für dieses Programm. Ein Paket mit einer PID von null, das heißt die Programmzugangstabelle **40**, liefert den Eintrittspunkt für jeglichen MPEG-Zugriff.

[0058] Um Anwendungen und Daten für sie herunterzuladen, werden zwei neue Stromtypen definiert, und die relevante PMT enthält außerdem Verweise auf die PIDs der Ströme von Anwendungs-MPEG-Tabellen **44** (oder Sektionen dieser) und Daten-MPEG-Tabellen **45** (oder Sektionen dieser). Obwohl es in bestimmten Fällen zweckmäßig sein kann, separate Stromtypen für ausführbare Anwendungssoftware und Daten zur Verarbeitung durch solche Software zu definieren, ist dies tatsächlich nicht wesentlich. Bei anderen Realisierungen kann man ausführbaren Code in einem einzigen Strom, auf den die PMT zugreift, wie beschrieben assemblieren.

[0059] Mit Bezug auf Fig. 4 wird, um zum Beispiel eine Anwendung in einem Strom **44** herunterzuladen, die Anwendung **46** in Module **47** aufgeteilt, die jeweils durch eine MPEG-Tabelle gebildet werden. Bestimmte dieser Tabellen umfassen eine einzige Sektion, während andere aus mehreren Sektionen **48** bestehen können. Eine typische Sektion **48** besitzt einen Kopfteil, der eine Ein-Byte-Tabellenidentifikation ("TID") **50**, die Sektionsnummer **51** dieser Sektion in der Tabelle, die Gesamtzahl **52** der Sektionen in dieser Tabelle und einen Erweiterungsverweis **53** von zwei Byte 110 enthält. Jede Sektion enthält außerdem einen Datenteil **54** und einen CRC **55**. Für eine bestimmte Tabelle **47** weisen alle Sektionen **48**, aus denen diese Tabelle **47** besteht, dieselbe TID **50** und dieselbe TID-Erweiterung **53** auf. Für eine bestimmte Anwendung **46** weisen alle Tabellen **47**, aus denen diese Anwendung **46** besteht, dieselbe TID **50**, aber verschiedene jeweilige TID-Erweiterungen auf.

[0060] Für jede Anwendung **46** wird eine einzige MPEG-Tabelle als eine Verzeichnistabelle **56** verwendet. Die Verzeichnistabelle **56** enthält in ihrem Kopfteil dieselbe TID wie die anderen Tabellen **47**, aus denen die Anwendung besteht. Die Verzeichnistabelle besitzt jedoch eine vorbestimmte TID-Erweiterung von null für Identifikationszwecke und aufgrund des Umstands, daß nur eine einzige Tabelle für die Informationen in dem Verzeichnis notwendig ist. Alle anderen Tabellen **47** besitzen normalerweise von null verschiedene TID-Erweiterungen und bestehen aus einer Anzahl assoziierter Sektionen **48**. Der Kopfteil der Verzeichnistabelle enthält außerdem eine Versionsnummer der herunterzuladenden Anwendung.

[0061] Wieder mit Bezug auf Fig. 3 werden die PMTs **40**, PMTs **41** und Anwendungs- und Datenstromkomponenten **44**, **45** zyklisch gesendet. Jede Anwendung, die gesendet wird, besitzt eine jeweilige vorbestimmte TID. Um eine Anwendung herunterzuladen, wird die MPEG-Tabelle mit der entsprechenden TID und einer TID-Erweiterung von null in den Empfänger/Decoder heruntergeladen. Diese ist die Verzeichnistabelle für die erforderliche Anwendung. Die Daten in diesem Verzeichnis werden dann durch den Decoder verarbeitet, um die TID-Erweiterungen der Tabellen, aus denen die erforderliche Anwendung besteht, zu bestimmen. Danach können eine etwaige erforderliche Tabelle mit derselben TID wie die Verzeichnistabelle und einer aus dem Verzeichnis bestimmten TID-Erweiterung heruntergeladen werden.

[0062] Der Decoder ist ausgelegt, die Verzeichnistabelle auf eine etwaige Aktualisierung dieser zu prüfen. Dies kann durch nochmaliges periodisches Herunterladen der Verzeichnistabelle, zum Beispiel alle 30 Sekunden oder jede Minute oder alle 5 Minuten, und Vergleichen der Versionsnummer der zuvor heruntergeladenen Verzeichnistabelle erfolgen. Wenn die frisch heruntergeladene Versionsnummer die einer späteren Version ist, werden die mit der vorherigen Verzeichnistabelle assoziierten Tabellen gelöscht und die mit der neuen Version assoziierten Tabellen heruntergeladen und assembliert.

[0063] Bei einer alternativen Anordnung wird der ankommende Bitstrom unter Verwendung einer der TID, der TID-Erweiterung und der Versionsnummer entsprechenden Maske mit für die TID der Anwendung, einer TID-Erweiterung von null und einer Versionsnummer von eins größer als die Versionsnummer des gerade heruntergeladenen Verzeichnisses gesetzten Werten gefiltert. Folglich kann eine Erhöhung der Versionsnummer erkannt werden, und nach der Erkennung wird das Verzeichnis heruntergeladen und die Anwendung aktualisiert, wie oben beschrieben. Wenn eine Anwendung beendet werden soll, wird ein leeres Verzeichnis mit der nächsten Versionsnummer gesendet, aber ohne jegliche in dem Verzeichnis aufgelistete Module. Als Reaktion auf den Empfang eines solchen leeren Verzeichnisses ist

der Decoder 2020 dafür programmiert, die Anwendung zu löschen.

[0064] In der Praxis können Software und Computerprogramme zur Implementierung von Anwendungen im Decoder über beliebige der Teile des Decoders eingeführt werden, insbesondere in dem, wie beschrieben, über die Satellitenstrecke empfangenen Datenstrom, aber auch über den seriellen Port, die Chipkartenstrecke usw. Solche Software kann Anwendungen auf hoher Ebene umfassen, mit denen interaktive Anwendungen in dem Decoder implementiert werden, wie zum Beispiel Net-Browser, Quiz-Anwendungen, Programmplaner usw. Es kann auch Software heruntergeladen werden, um die Arbeitskonfiguration der Decodersoftware zu ändern, zum Beispiel mittels "Patches" oder dergleichen.

[0065] Anwendungen können auch über den Decoder heruntergeladen und zu einem PC oder dergleichen, der mit dem Decoder verbunden ist, gesendet werden. In einem solchen Fall wirkt der Decoder als Kommunikationsrouter für die Software, die letztendlich auf der angeschlossenen Einrichtung abläuft. Zusätzlich zu dieser Routing-Funktion kann der Decoder auch zum Umsetzen der paketierten MPEG-Daten vor dem Routen zu dem PC zu Computerdateisoftware fungieren, die zum Beispiel gemäß dem DSMCC-Protokoll organisiert ist (siehe unten).

[0066] Zuvor konzentrierten sich zum Verifizieren der Vollständigkeit des Ursprungs von Anwendungsdaten implementierte Maßnahmen auf das Verifizieren der Tabellen in dem MPEG-Paketstrom. Insbesondere wird bei herkömmlichen Systemen eine Hash-Funktion vor der Übertragung auf jede der einzelnen Sektionen **48** angewandt und der resultierende Prüfwert bzw. die Signatur für jede Sektion wird in einer Liste in der Verzeichnungstabelle **56** gespeichert, die zu dem Decoder gesendet wird. Ein Vergleich des danach durch den Decoder berechneten Hash-Werts mit dem in dem Verzeichnis für eine empfangene Sektion gespeicherten Prüfwerts ermöglicht das Verifizieren der Integrität der empfangenen Sektion.

[0067] Daten in dem Verzeichnis **40** können gleichermaßen einem Hash-Prozeß unterzogen werden, um einen weiteren Prüfwert bzw. eine Signatur für die Verzeichnistabelle **40** zu erzeugen. Ferner kann dieser Prüfwert durch einen privaten Schlüssel verschlüsselt und in der Verzeichnistabelle gespeichert werden. Nur die Decoder, die einen entsprechenden öffentlichen Schlüssel besitzen, können die Signatur authentifizieren.

[0068] Im Gegensatz zu solchen herkömmlichen Systemen betrifft die vorliegende Ausführungsform ein Mittel zum Sichern und Verifizieren von in einer Mehrfach-Hierarchie von Daten-Dateien oder Objekten organisierten Anwendungsdaten auf der Ebene der Anwendung. Dies wird aus Fig. 5, die die Beziehung zwischen den in einer Menge von DSM-CC-U-U-Daten-Dateien **60** organisierten Daten in einer assemblierten Anwendung **46** und in einer Reihe von MPEG-Tabellen **47** eingekapselt zeigt.

[0069] Vor der Übertragung werden die Daten-Dateien zu der Anwendung **46** assembliert und danach durch den MPEG-Kompressor zu MPEG-Tabellen oder -modulen **47** wie oben beschrieben formatiert, wobei ein für den MPEG-Paketstrom spezifischer Kopfteil **49**, der die Tabellen-ID, Versionsnummer usw. enthält, mit eingeschlossen wird. Diese Tabellen werden dann durch den MPEG-Kompressor in MPEG-Pakete eingekapselt. Es versteht sich, daß möglicherweise keine feste Beziehung zwischen den in den Daten-Dateien **61** organisierten Daten und den letztendlichen MPEG-Tabellen **47** besteht. Nach Empfang und Filterung durch den Decoder werden die Paketkopfteile verworfen und die Reihe von Tabellen wird aus den Nutzinformationen der ausgestrahlten Pakete rekonstituiert. Danach werden die Tabellenkopfteile **49** verworfen und die Anwendung **46** aus den Nutzinformationen der Tabellen **47** rekonstituiert.

[0070] Das DSMCC-Format für Daten-Dateien ist eine insbesondere für die Verwendung in Multimedia-Netzwerken ausgelegte Norm, die eine Reihe von Nachrichtenformaten und Sitzungsbefehlen zur Kommunikation zwischen einem Client-Benutzer **70**, einem Server-Benutzer **71** und einem Netzwerkbetriebsmittelmanager **72** definiert. Siehe Fig. 6. Der Netzwerkbetriebsmittelmanager **72** kann als logische Entität betrachtet werden, die zum Verwalten der Zuordnung von Betriebsmitteln in einem Netzwerk wirkt. Obwohl sie anfänglich für die Verwendung im Kontext der bidirektionalen Netzwerkkommunikation konzipiert wurden, konzentrierten sich neuere Implementierungen der Norm DSM-CC auf ihre Verwendung für unidirektionale Ausstrahlungszwecke.

[0071] Die Kommunikation zwischen einem Client und einem Server wird durch eine Reihe von Sitzungen aufgebaut, wobei eine erste Reihe von Nachrichten zwischen einem Benutzer (Client **70** oder Server **71**) und dem Netzwerkmanager **72** ausgetauscht wird, um den Client und/oder den Server für Kommunikation zu konfigurieren. Solche Nachrichten werden gemäß dem sogenannten DSMCC-U-N-Protokoll (Benutzer zu Netzwerk) formatiert. Eine Teilmenge dieses Protokolls wurde insbesondere zum Ausstrahlungs-Herunterladen von Daten definiert.

[0072] Nachdem eine Kommunikationsstrecke hergestellt wurde, werden nachfolgend Nachrichten zwischen Client **70** und Server **71** gemäß dem DSM-CC-U-U (Protokoll von Benutzer zu Benutzer) ausgetauscht. Eine Sequenz von Nachrichten dieser Art

entspricht den Daten-Dateien **60** von Fig. 5. Im Fall von DSMCC-U-U-Nachrichten werden Daten zu einer Reihe von Nachrichten **61** organisiert, die gemäß dem Protokoll BIOP (Broadcast InterOrb) gruppiert werden.

[0073] Jede Nachricht bzw. jedes Objekt **61** umfaßt einen Kopfteil **62**, einen Sub-Kopfteil **63** und Nutzinformationen **64**, die die Daten selbst enthalten. Gemäß dem BIOP-Protokoll enthält der Kopfteil **62** u.a. eine Anzeige des Typs der Nachricht und der BIOP-Version, während der Sub-Kopfteil die Art des Objekts und andere, durch den Systemarchitekt zu definierende Informationen angibt.

[0074] Datenobjekte **64** in den Nutzinformationen von DSMCC-U-U-Dateien können im allgemeinen als einer von drei Typen definiert werden: Verzeichnisobjekte, Dateiobjekte und Stromobjekte. Verzeichnisobjekte definieren Wurzelverzeichnisse oder Unterverzeichnisse zum Verweis auf eine Reihe von assoziierten Dateiobjekten, die die tatsächlichen Anwendungsdaten enthalten.

[0075] Stromobjekte können verwendet werden, um das Herstellen einer zeitlichen Beziehung zwischen in den Daten-Dateien enthaltenen Daten und dem MPEG-Paketstrom selbst zu ermöglichen. Man kann dies zum Beispiel im Fall von interaktiven Anwendungen verwenden, die in den Daten-Dateien enthalten und dafür ausgelegt sind, mit den durch den Decoder empfangenen und verarbeiteten elementaren Video- oder Audioströmen synchronisiert zu werden. Wie bereits erwähnt, kann andernfalls keine direkte Korrelation zwischen den paketierten MPEG-Daten und den Daten-Dateien bestehen.

[0076] Im Gegensatz zu den MPEG-Tabellen, bei denen ein einziges Verzeichnis eine Menge von Tabellen mit nur einer einzigen Hierarchieebene referenziert, können die Daten-Dateien **60** auf relativ komplexere hierarchische Weise organisiert werden. Wie bei in einem PC oder Server gespeicherten Dateien kann ein Haupt- oder Wurzelverzeichnis auf eines oder mehrere Unterverzeichnisse verweisen, die ihrerseits auf eine zweite Ebene von Daten-Dateien verweisen. Man kann sogar auf ein mit einer weiteren Menge von Anwendungsdaten assoziiertes zweites Wurzelverzeichnis Bezug nehmen.

[0077] Mit Bezug auf Fig. 7 ist ein Beispiel für eine Dateistruktur für eine Menge von Daten-Dateien oder -Einheiten gezeigt. Ein als **75** bezeichnetes Wurzelverzeichnis DIR A0 referenziert eine Gruppe von Unterverzeichnissen A1 bis A4 mit der Bezeichnung **76**. Jedes Unterverzeichnis **76** referenziert eine oder mehrere Mengen assoziierter Objektdateien **77**. Der Klarheit halber ist nur eine einzige Gruppe von Objektdateien F1, F2 usw., die mit dem Unterverzeichnis A4 assoziiert ist, gezeigt. In der Praxis kann jedes der Unterverzeichnisse A1 bis A4 eine Anzahl von Gruppen oder Objektdateien referenzieren.

[0078] In jedem Verzeichnis und Unterverzeichnis wird eine Menge von Authentifikationsschritten für die mit diesem Verzeichnis verknüpften 20 Dateien eingeführt. Mit Bezug auf das Wurzelverzeichnis **75** umfaßt der Sub-Kopfteil **63** einen Hash-Wert, der durch Anwenden eines Hash-Algorithmus auf bestimmte oder alle der in den Unterverzeichnisdateien A1 bis A4 mit der Kennzeichnung **76** gespeicherten Daten erhalten wird. Der Hash-Algorithmus kann von einem beliebigen bekannten Typ sein, wie zum Beispiel MD5 (Message Digest algorithm).

[0079] Bei einer Realisierung kann der Algorithmus vor der Übertragung auf jede assoziierte Datei oder jedes assoziierte Unterverzeichnis einzeln angewandt und eine Liste der Hash-Werte für jedes Unterverzeichnis **76** in dem Wurzelverzeichnis **75** gespeichert werden. Obwohl eine solche Lösung einen vergrößerten Prüfauflösungsgrad im Hinblick auf das Verifizieren jedes Unterverzeichnisses ermöglicht, kann diese Lösung jedoch im Hinblick auf die Verarbeitungszeit, die der Decoder zum Berechnen der entsprechenden Signaturen benötigt, ineffizient sein.

[0080] Der Sub-Kopfteil **63** des Verzeichnisses **79** umfaßt folglich vorzugsweise einen kumulativen Flash-Wert **79**, der durch Anwenden des MD5-Hash-Algorithmus auf die kombinierten Sub-Kopfteil- und Nutzinformationssektionen **63**, **64** der Unterverzeichnisse **76**, das heißt, ohne den Kopfteil **62**, berechnet wird. Insbesondere werden die in den Unterverzeichnissen **76** enthaltenen und auf die Schicht von Dateiobjekten **77** verweisenden Flash-Werte **82** in diese Hash-Berechnung aufgenommen.

[0081] Im Fall des in Fig. 7 gezeigten Unterverzeichnisses A4 verweist dieses Unterverzeichnis selbst auf eine Menge von Objektdateien F1–Fn, die bei **77** angegeben ist. In diesem Fall wird ein kumulativer Flash-Wert **82** für die kombinierten Inhalte der Objektdateien **77** erzeugt. Dieser Wert wird in den Hash-Prozeß aufgenommen, der zu dem Flash-Wert **79** führt. Es ist deshalb nicht möglich, irgendwelche der Objektdateien **77** zu ändern, ohne den Flash-Wert **82** des Unterverzeichnisses **76** zu ändern, wodurch wiederum der Flash-Wert **79** des Verzeichnisses **75** verändert wird.

[0082] Im vorliegenden Fall wird für alle Unterverzeichnisse A1–A4, die in dem Verzeichnis referenziert werden, ein kombinierter Hash-Wert berechnet. Dieser Hash-Wert wird zusammen mit einer Kennung der Gruppe von Unterverzeichnissen, aus denen die Daten entnommen werden, gespeichert. Bei anderen Ausführungsformen kann eine Reihe kombinierter oder einzelner Hash-Werte und entsprechende Ken-

nungen in dem Sub-Kopfteil des Verzeichnisses gespeichert werden.

[0083] Zum Beispiel kann auch eine zweite Menge von Unterverzeichnissen, die auch mit dem Wurzelverzeichnis assoziiert ist, aber eine andere Menge von Daten oder ausführbarem Code betrifft, miteinander gruppiert und ein kumulativer Hash-Wert für diese Unterverzeichnisse berechnet und in dem Sub-Kopfteil-Wurzelverzeichnis gespeichert werden. Gleichermaßen kann ein einziger, mit einem einzigen Verzeichnis assoziierter Hash-Wert in dem Sub-Kopfteil des Wurzelverzeichnisses gespeichert werden.

[0084] Die Authorisation von Gruppen oder individuellen Daten-Dateien schließt natürlich nicht aus, daß das Wurzelverzeichnis (oder 25 tatsächlich irgendeine andere Datei) auch auf nichtvalidierte oder ungehashte Daten-Dateien verweist, das Fehlen der Validierung einer solchen Datei muß jedoch bei etwaigen Operationen mit dieser Datei berücksichtigt werden. In dieser Hinsicht ist es möglicherweise zum Beispiel nicht notwendig, Stromobjekte zu authentifizieren.

[0085] Die Verwendung einer Hash-Funktion ermöglicht es in diesem Fall hauptsächlich dem Decoder, die Integrität oder Vollständigkeit der heruntergeladenen Daten-Dateien zu verifizieren. Zum Beispiel im Fall eines Fehlers oder einer Unterbrechung der Übertragung ergibt die Operation eines kumulativen Hash-Algorithmus an den empfangenen abhängigen Dateien nicht dasselbe Ergebnis wie der in dem Wurzelverzeichnis gespeicherte Hash-Wert für diese Dateien.

[0086] Der Decoder wird dann auf die Anwesenheit möglicher Fehler in den heruntergeladenen Daten hingewiesen und lädt die fehlerhaften Daten-Dateien neu.

[0087] Es versteht sich, daß im Fall eines Hash-Algorithmus die Berechnung des Hash-Werts gemäß einer öffentlich bekannten Reihe von Berechnungsschritten ausgeführt wird und folglich beliebige Personen den Hash-Wert für eine gegebene Menge von Daten-Dateien erzeugen können. Es ist also normalerweise nicht möglich, durch einfaches Prüfen der Hash-Werte den Ursprung solcher Daten-Dateien zu verifizieren.

[0088] Um dieses Problem zu überwinden, wird ein Signaturwert für das Wurzelverzeichnis **75** unter Verwendung eines nur dem Betreiber bekannten geheimen Schlüsselwerts berechnet. Dieser Schlüssel kann einem durch einen symmetrischen Schlüsselalgorithmus, wie zum Beispiel den DES-Algorithmus (Data Encryption Standard), erhaltenen Schlüssel entsprechen. Es wird aber bevorzugt ein Algorithmus mit privaten/öffentlichen Schlüsseln, wie zum Beispiel der RSA-Algorithmus (Rivest, Shamir und Adelman) verwendet, wobei der für das Erzeugen der Daten-Dateien verantwortliche Betreiber den privaten Schlüsselwert besitzt und die öffentlichen Schlüsselwerte von den Decodern gehalten werden.

[0089] Wie in Fig. 7 gezeigt, umfaßt das Wurzelverzeichnis **75** eine Schlüsselkennung oder magische Nummer **80**, die dem Decoder den in der Verifikationsphase zu verwendenden öffentlichen Schlüssel zusammen mit dem berechneten Signaturwert **81**, der unter Verwendung des privaten Schlüssels des Betreibers erzeugt wird, identifiziert. In diesem Fall wird der Signaturwert **81** durch Anwenden des vom Betreiber gehaltenen privaten Schlüssels auf bestimmte oder alle der Daten in dem Verzeichnis **75**, vorzugsweise einschließlich der Nutzinformationsdaten **64** und/oder des kumulativen Hash-Werts bzw. der kumulativen Hash-Werte **79** erzeugt. Der Decoder kann diesen Signaturwert **81** dann unter Verwendung des durch die Schlüsselnummer **80** identifizierten entsprechenden öffentlichen Schlüssels verifizieren.

[0090] In diesem Beispiel sind die Daten in dem Verzeichnis **75** unverschlüsselt und der private Schlüssel dient lediglich zur Bereitstellung eines durch den öffentlichen Schlüssel verifizierbaren Signaturwerts. Bei alternativen Ausführungsformen können bestimmte oder alle der Inhalte des Verzeichnisses durch den privaten Schlüssel verschlüsselt und danach durch einen entsprechenden Schlüssel entschlüsselt werden.

[0091] In jedem Fall ermöglicht es die Erzeugung eines Signaturwerts oder Blocks verschlüsselten Codes durch Verwendung eines geheimen Schlüssels einen Decoder, die Integrität und den Ursprung des Verzeichnisses **75** und durch Implikation die Integrität und den Ursprung der Dateien, auf die dieses Wurzelverzeichnis verweist, zu verifizieren. Da die kumulativen Hash-Werte für die verwiesenen Dateien in der Berechnung der Signatur **81** enthalten sind, ist es nicht möglich, diese Werte zu verändern, ohne daß dies in der Verifikationsphase erkannt wird. Da jeder Hash-Wert im allgemeinen für eine gegebene Menge von Daten einzigartig ist, wäre es deshalb nicht möglich, den Inhalt irgendwelcher der abhängigen gehashten Dateien zu ändern, ohne ihren charakteristischen Hash-Wert und dadurch den resultierenden Signaturwert eines Verzeichnisses zu ändern.

[0092] Das Wurzelverzeichnis **75**, Unterverzeichnisse **76** und Objektdateien **77** werden alle durch einen Ausstrahlungsbetreiber des Systems, der hier als Betreiber A angegeben ist, erzeugt. In diesem Fall besitzen diese Dateien alle einen bekannten und verifizierbaren gemeinsamen Ursprung.

[0093] Abhängig von der zu implementierenden Anwendung kann jedoch gleichermaßen auf eine Men-

ge von mit einem zweiten Betreiber B assoziierten Daten-Dateien Bezug genommen werden. In diesem Fall enthält das Unterverzeichnis **76** einen Verweis auf das Wurzelverzeichnis DIR B0 einer zweiten Menge von Daten-Dateien, die bei **78** angegeben ist. Es ist auch möglich, Verbindungen zwischen Daten-Dateien von verschiedenen Quellen auf anderen Ebenen, wie zum Beispiel einer Dateihierarchie, bei der ein erstes Unterverzeichnis in einer Menge von Dateien auf ein Unterverzeichnis einer zweiten Menge von Daten-Dateien verweist usw., in Betracht zu ziehen.

[0094] Wie bei dem Wurzelverzeichnis DIR A0 für den Betreiber A enthält das bei **78** angegebene Wurzelverzeichnis DIR B0 einen oder mehrere kumulative Flash-Codewerte **84**, die mit ihren assoziierten (nicht gezeigten) Unterverzeichnissen assoziiert sind, einer Schlüsselnummer **84**, die den öffentlichen Schlüssel des Betreibers B identifiziert, der im Verifizierungsschritt verwendet werden soll, und einen durch den entsprechenden Betreiber-Privatschlüssel erzeugten Signaturwert **86**.

[0095] Unter Verwendung des Flash-Werts **84** und der Signatur **86** in dem Sub-Kopfteil des Verzeichnisses und auch der Nutzinformationsdaten **64** des Verzeichnisses **78** wird ein Hash-Wert für dieses Verzeichnis berechnet. Dieser Hash-Wert wird dann in dem Unterverzeichnis A4 gespeichert, wodurch eine Verifikation der Integrität der Daten in der Verzeichnistabelle ausgeführt werden kann.

[0096] Aufgrund des Umstands, daß die Signatur **86** und die Hash-Werte **84** in der Berechnung des Hash-Werts **83** enthalten sind, kann auch die Integrität des Rests der Daten-Dateien, auf die das Wurzelverzeichnis **78** verweist, angenommen werden, da keine dieser abhängigen Dateien geändert werden kann, ohne den Hash-Wert **84** zu ändern und insbesondere den Signaturwert **86** zu ändern. Da der Signaturwert **86** nur durch eine Person berechnet werden kann, die den privaten Betreiberschlüssel besitzt, kann die Integrität aller Dateien, auf die das Verzeichnis **78** verweist, angenommen werden, solange entsprechende Hash-Werte für weitere abhängige Unterverzeichnisse und Objektdateien berechnet werden.

[0097] Auf diese Weise können Anwendungsdaten in bezug auf ausführbare Programme oder dergleichen, die von einem zweiten Betreiber erzeugt werden, auf sichere und zuverlässige Weise mit Anwendungen verknüpft werden, die mit einem ersten Betreiber assoziiert sind.

[0098] Es versteht sich, daß eine Anzahl von Varianten möglich sind, insbesondere zur Reduktion der Menge an in jeder Phase gehashten oder signierten Daten. Insbesondere können im Fall einer Signatur oder eines Hash-Werts in einem Verzeichnis oder Unterverzeichnis, mit dem eine Daten-Datei auf einer niedrigeren Ebene verifiziert wird, die Verzeichnissignatur oder der Hash-Wert unter Verwendung nur des Hash-Werts der niedrigeren Ebene und keine anderen Daten erzeugt werden.

[0099] Zum Beispiel kann der kombinierte Hash-Wert **79** in dem A0-Verzeichnis **75** unter Verwendung der kombinierten Hash-Werte **82**, **83** jeder der bei **76** angegebenen Unterverzeichnisse A1–A4 erzeugt werden. Da diese Werte genauso einzigartig wie die Daten in den Nutzinformationen des Unterverzeichnisses sind, ist der kombinierte Hash-Wert **79** immer noch einzigartig für die betreffenden Unterverzeichnisse. Weiterhin kann immer noch die Integrität der niedrigeren Ebene von Objekt- und Verzeichnisdateien **77**, **78** angenommen werden, da die Hash-Werte **82** weiterhin in der Berechnung verwendet werden.

[0100] Gleichermaßen kann man unter Verwendung des Signaturwerts **86** den Hash-Wert **82** berechnen, der zum Verifizieren des bei **78** angegebenen B0-Verzeichnisses berechnet wird. Da dies von den Hash-Werten **84** abhängt und eindeutig mit ihnen assoziiert ist, wobei diese Hash-Werte ihrerseits von der nächsten Ebene von Dateien abhängen, kann immer noch die Integrität der Gesamtheit der Mengen von Daten-Dateien, auf die das Verzeichnis **78** verweist, angenommen werden.

Patentansprüche

1. Vorrichtung (**2**) zum Authentifizieren von in einem digitalen Übertragungssystem gesendeten Daten, wobei die Vorrichtung folgendes umfaßt:
ein Mittel (**6**) zum Organisieren der Daten vor der Übertragung zu einer Hierarchie aus mindestens einer Wurzelverzeichniseinheit, einer Unterverzeichniseinheit und einer Dateieinheit,
wobei die Vorrichtung **dadurch gekennzeichnet** ist, daß sie ferner folgendes umfaßt:
ein Mittel (**6**) zum Wirken an Daten in einer Datei durch einen ersten Authentifikationsalgorithmus,
ein Mittel (**6**) zum Speichern eines assoziierten Dateiauthentifikationswerts in dem verweisenden Unterverzeichnis,
ein Mittel (**6**), das seinerseits durch einen zweiten Authentifikationsalgorithmus an diesem Dateiauthentifikationswert wirkt, und
ein Mittel (**6**) zum Speichern eines assoziierten Unterverzeichnisses-Authentifikationswerts in dem verweisenden Wurzelverzeichnis.

2. Vorrichtung nach Anspruch 1, ferner dadurch gekennzeichnet, daß der erste Authentifikationsalgorithmus ein RSA-Algorithmus ist.

3. Vorrichtung nach Anspruch 1, ferner dadurch

gekennzeichnet, daß der zweite Authentifikationsalgorithmus ein RSA-Algorithmus ist.

4. Vorrichtung nach Anspruch 1, ferner dadurch gekennzeichnet, daß der erste Authentifikationsalgorithmus einen MD5- oder einen SHA-Algorithmus umfaßt.

5. Vorrichtung nach Anspruch 1, ferner dadurch gekennzeichnet, daß der zweite Authentifikationsalgorithmus einen MD5- oder einen SHA-Algorithmus umfaßt.

6. Vorrichtung nach einem der vorherigen Ansprüche, bei der die Einheiten einer Menge von Daten-Dateien entsprechen, wobei die Daten-Dateien in Datentabellen oder -sektionen eingekapselt sind, wobei die Vorrichtung ferner dafür ausgelegt ist, die Tabellen oder Sektionen in Datenpakete einzukapseln, um einen Transportstrom zu bilden.

7. Vorrichtung nach Anspruch 6, bei der die Einheiten Datenobjekten entsprechen, die gemäß der Norm DSM-CC formatiert sind.

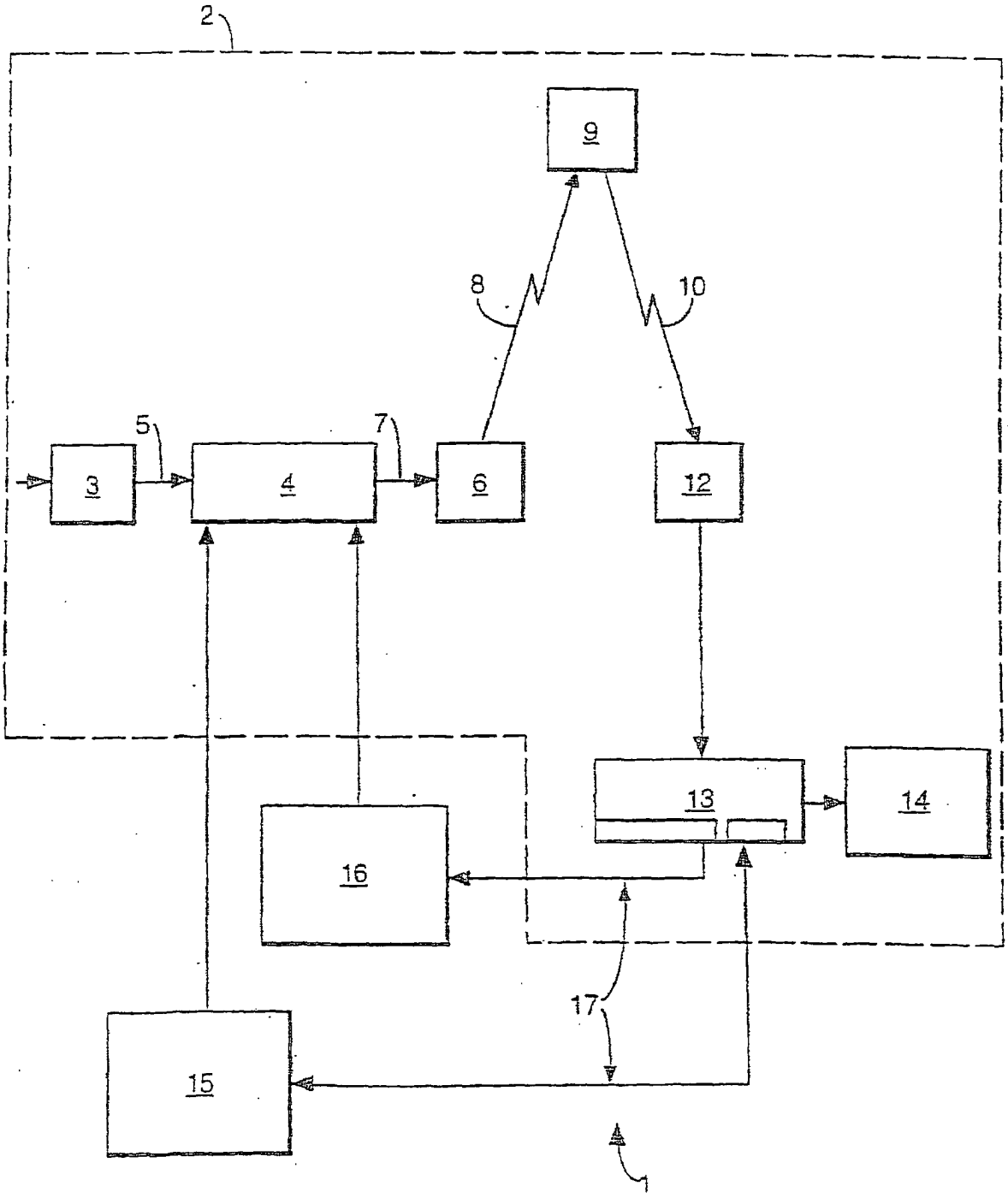
8. Vorrichtung nach Anspruch 6 oder 7, bei der die Einheiten in Tabellen und Pakete eingekapselt werden, die der MPEG-Norm entsprechen.

9. Digitales Signal, das Daten umfaßt, die aus mindestens einer Wurzelverzeichniseinheit (**75**), einer Unterverzeichniseinheit (**76**) und einer Dateieinheit (**77**) organisiert sind, dadurch gekennzeichnet, daß das Signal einen Dateiauthentifikationswert (**82**, **87**) umfaßt, der sich aus einem an einer Daten-Datei wirkenden ersten Authentifikationsalgorithmus ergibt, wobei der Dateiauthentifikationswert in dem verweisenden Unterverzeichnis gespeichert wird und sich aus dem Wirken eines zweiten Authentifikationsalgorithmus an dem Dateiauthentifikationswert ein Unterverzeichnis-Authentifikationswert (**63**) ergibt, wobei der Unterverzeichnis-Authentifikationswert in der verweisenden Wurzelverzeichniseinheit gespeichert wird.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

Fig.1.



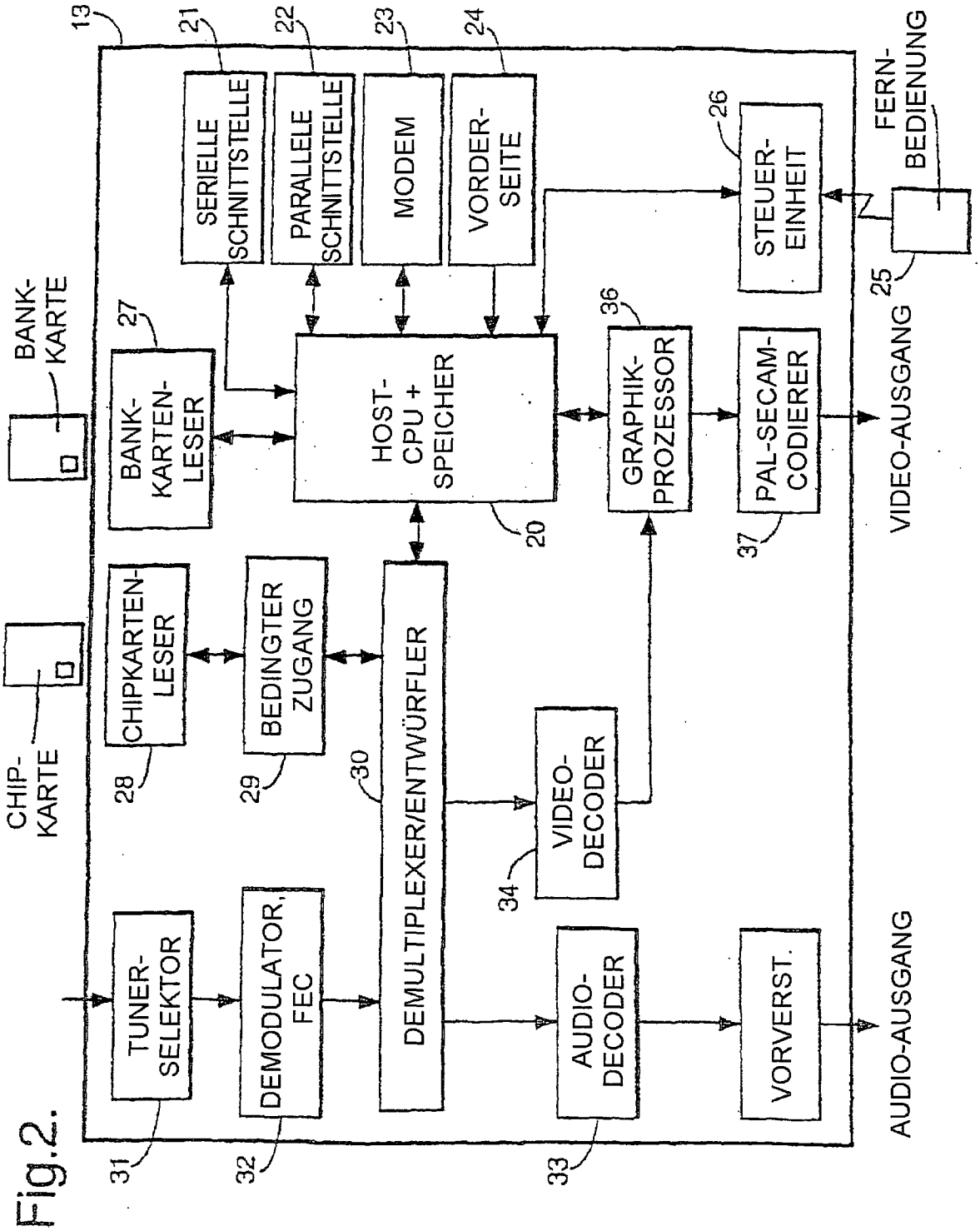

13
21
22
23
24
SERIELLE SCHNITTSTELLE
PARALLELE SCHNITTSTELLE
MODEM
VORDER-SEITE
26
STEUER-EINHEIT
FERN-BEDIENUNG
25
BANK-KARTE
27
BANK-KARTEN-LESER
HOST-CPU + SPEICHER
20
36
GRAPHIK-PROZESSOR
PAL-SECAM-CODIERER
37
VIDEO-AUSGANG
CHIP-KARTE
CHIPKARTEN-LESER
28
BEDINGTER ZUGANG
29
30
DEMULTIPLEXER/ENTWÜRFLER
VIDEO-DECODER
34
TUNER-SELEKTOR
31
DEMODULATOR, FEC
32
AUDIO-DECODER
33
VORVERST.
AUDIO-AUSGANG

Fig.2.

Fig.3.

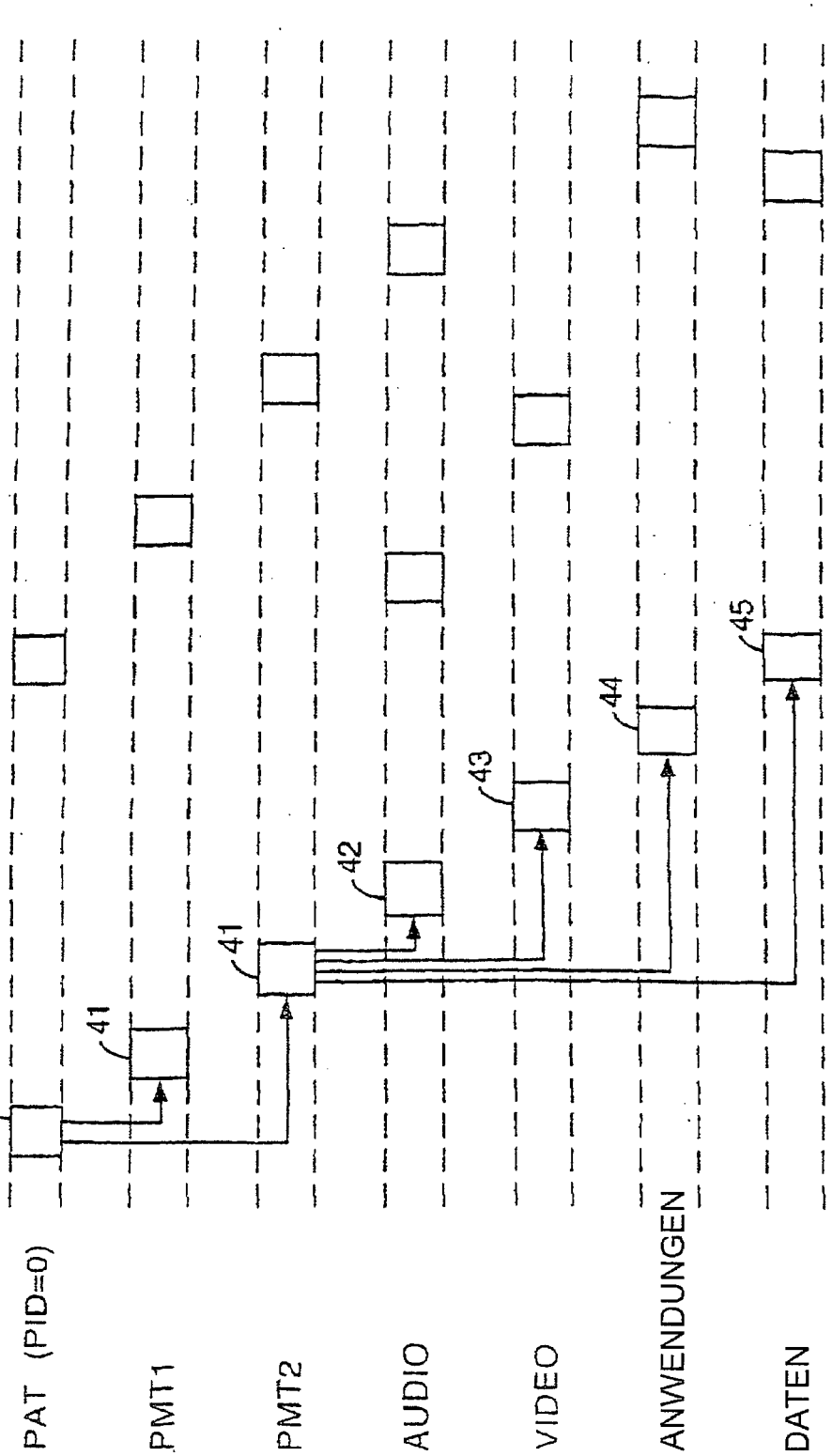

PAT (PID=0)
PMT1
PMT2
AUDIO
VIDEO
ANWENDUNGEN
DATEN
40
41
41
42
43
44
45

Fig.4.

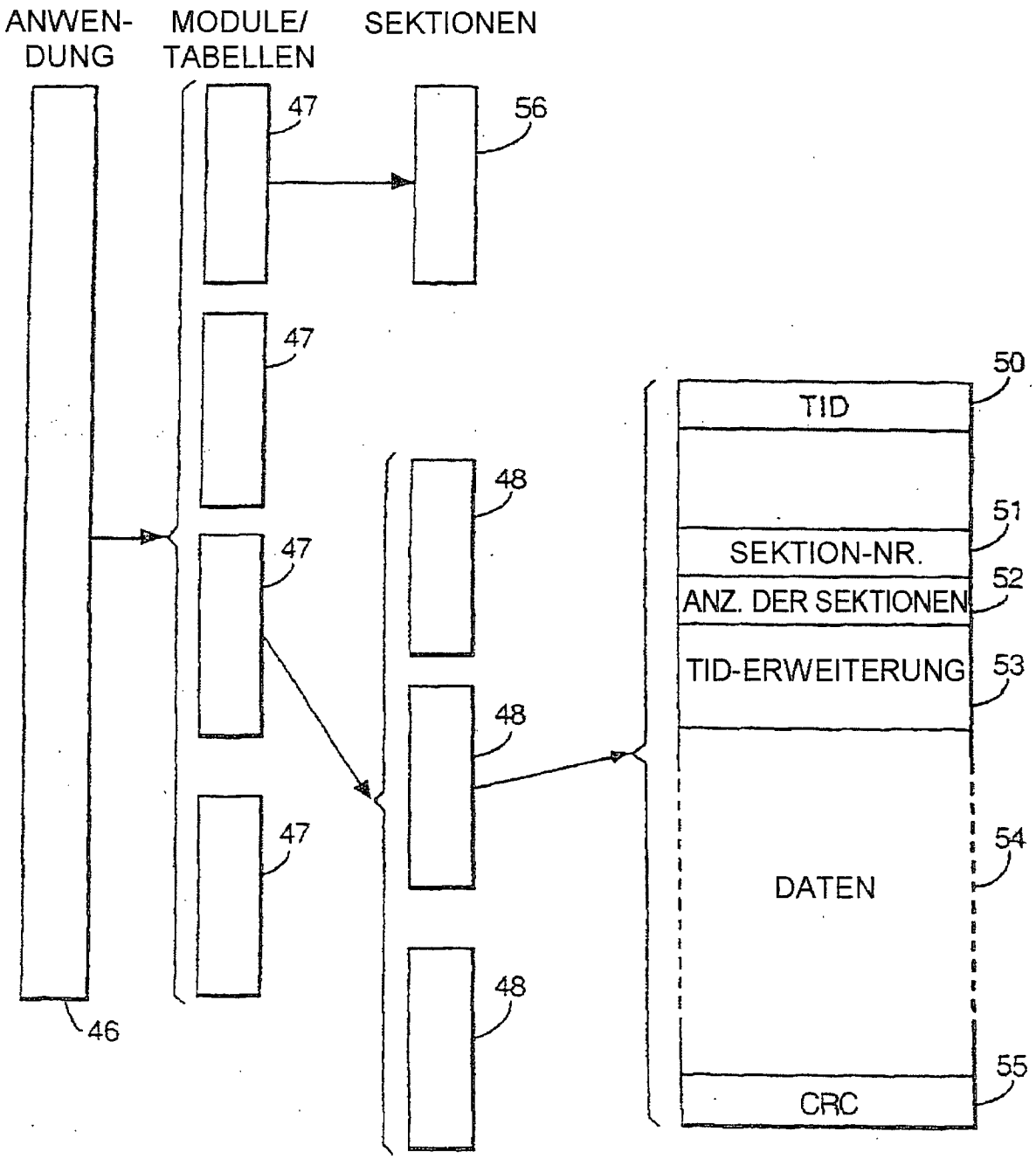
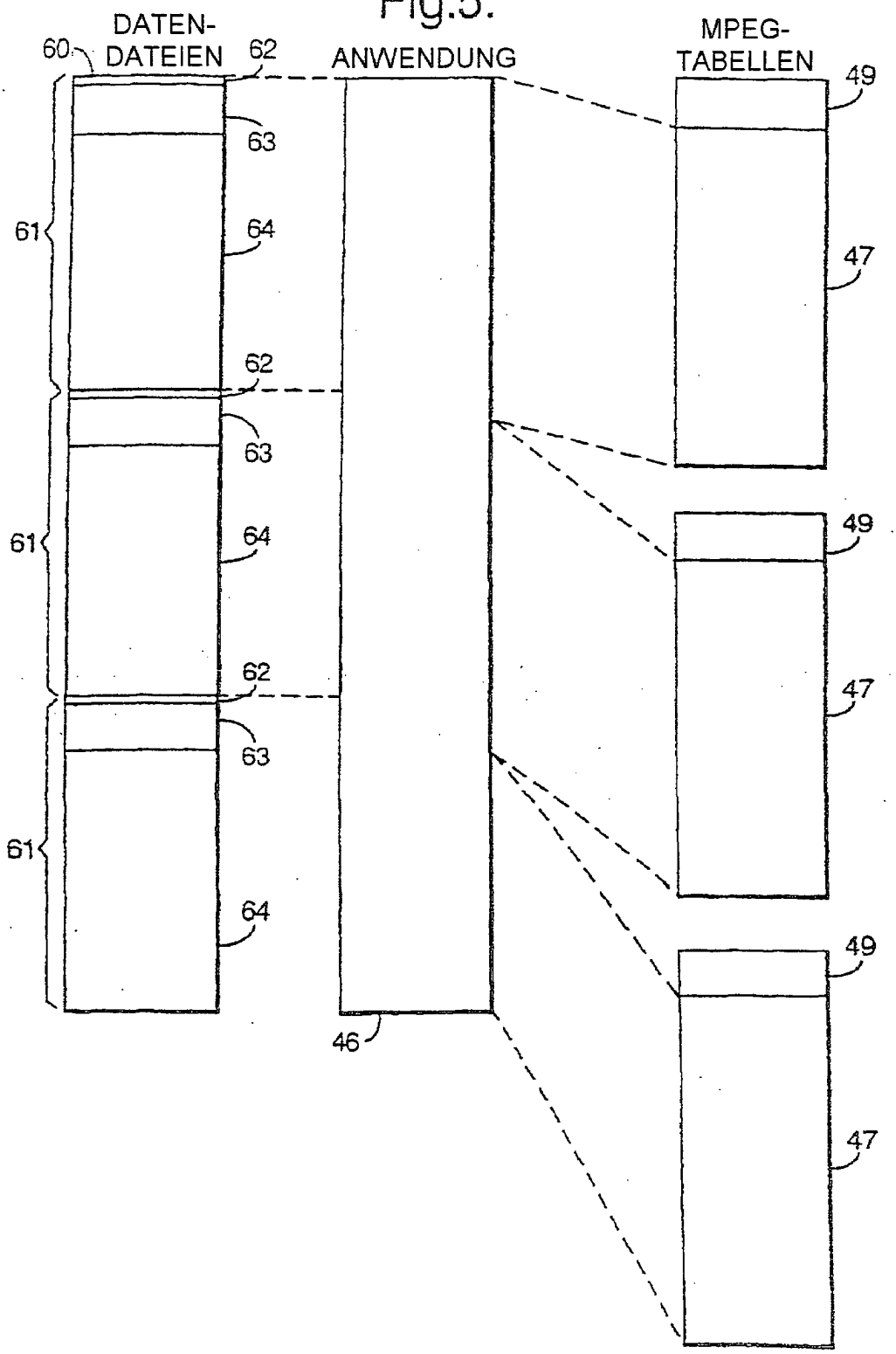

ANWEN-
DUNG
MODULE/
TABELLEN
SEKTIONEN
47
56
47
47
48
47
48
47
48
46
50
TID
51
SEKTION-NR.
52
ANZ. DER SEKTIONEN
53
TID-ERWEITERUNG
54
DATEN
55
CRC

Fig.5.


DATEN-
DATEIEN
60
62
ANWENDUNG
MPEG-
TABELLEN
49
63
64
47
61
62
63
64
49
47
61
62
63
64
49
47
61
46

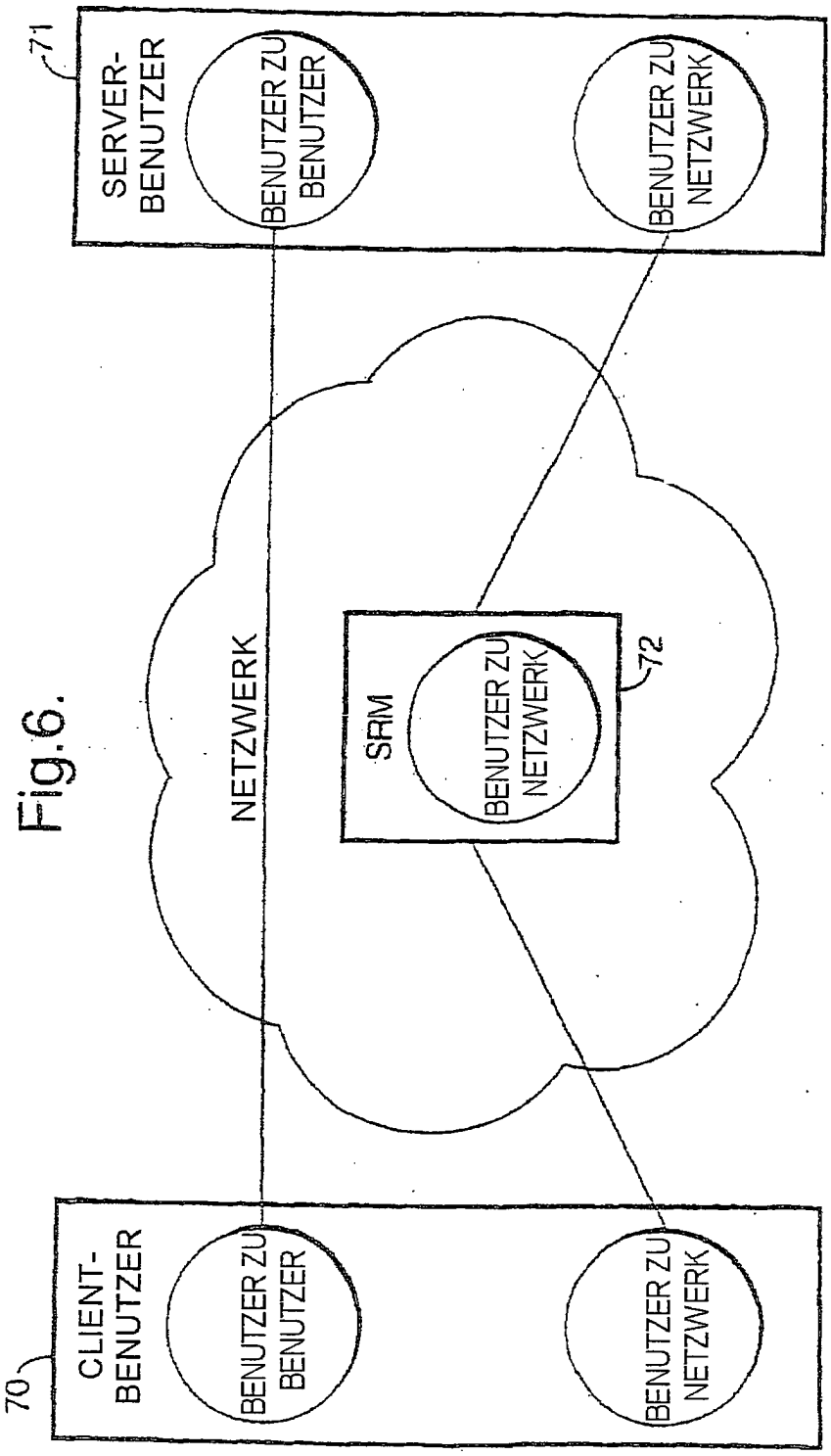

71
SERVER-BENUTZER
BENUTZER ZU BENUTZER
BENUTZER ZU NETZWERK
NETZWERK
SRM
BENUTZER ZU NETZWERK
72
70
CLIENT-BENUTZER
BENUTZER ZU BENUTZER
BENUTZER ZU NETZWERK

Fig.6.

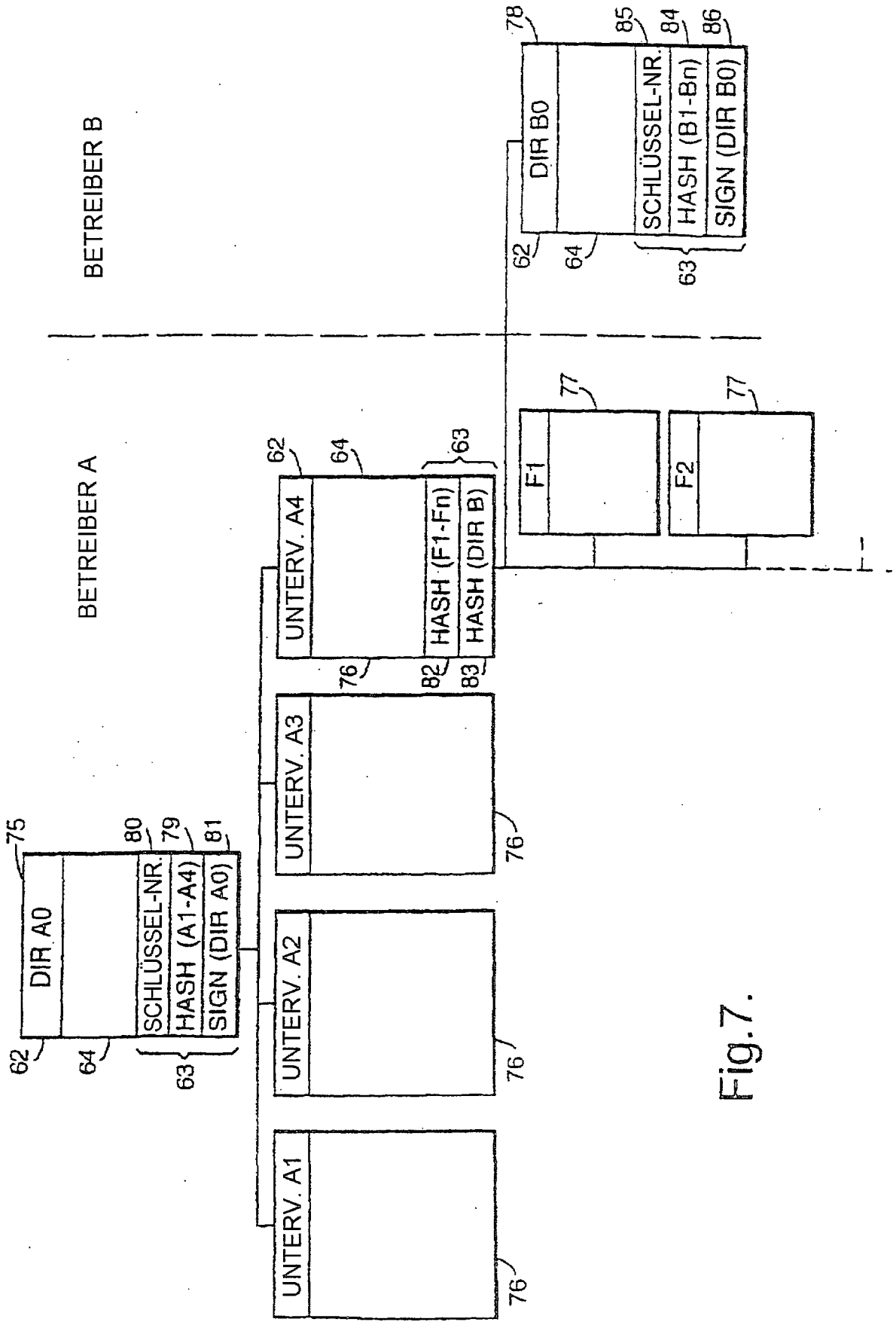

BETREIBER B
BETREIBER A
DIR B0
SCHLÜSSEL-NR.
HASH (B1-Bn)
SIGN (DIR B0)
F1
F2
UNTERV. A4
HASH (F1-Fn)
HASH (DIR B)
UNTERV. A3
UNTERV. A2
UNTERV. A1
DIR A0
SCHLÜSSEL-NR.
HASH (A1-A4)
SIGN (DIR A0)
62
63
64
75
76
77
78
79
80
81
82
83
84
85
86

Fig.7.